



Edge-centric inferential modeling & analytics

Christos Anagnostopoulos

School of Computing Science, University of Glasgow, UK

ARTICLE INFO

Keywords:

Edge inferential analytics

Quality of analytics

Communication efficiency

Hetero-associative learning vector quantization

Optimal stopping theory

ABSTRACT

This work contributes to a real-time, edge-centric inferential modeling and analytics methodology introducing the fundamental mechanisms for (i) predictive models update and (ii) diverse models selection in distributed computing. Our objective in edge-centric analytics is the time-optimized model caching and selective forwarding at the network edge adopting optimal stopping theory, where communication overhead is significantly reduced as only inferred knowledge and sufficient statistics are delivered instead of raw data obtaining high quality of analytics. Novel model selection algorithms are introduced to fuse the inherent models' diversity over distributed edge nodes to support inferential analytics tasks to end-users/analysts, and applications in real-time. We provide statistical learning modeling and establish the corresponding mathematical analyses of our mechanisms along with comprehensive performance and comparative assessment using real data from different domains and showing its benefits in edge computing.

1. Introduction

Real-time inferential analytics (Lazerson et al., 2016; Cormode, 2013) support exploratory (*hypotheses formulation*), diagnostic (*why is it happening*), predictive (*when is likely to happen*) and descriptive (*what is happening now*) data analysis via predictive statistical models e.g., multivariate linear & quartile regression over live data (Renart et al., 2017). The derived and incrementally updated predictive models, mainly regression and time-series forecasting models, are used in such analyses supporting analysts/applications in terms of: (i) real-time prediction of new/unseen data (regression) (ii) investigation how observed data fit such models (function estimation) and (iii) forecasting of future data trends of incoming data (Renart et al., 2017).

Real-time inferential analytics are materialized *after* contextual data are transferred from sensing devices and data sources to the Cloud aiming to build global on-line models over *all* observed data (Konečný et al., 2016). Then, analysts/applications issue arbitrary *regression & exploratory queries* over such models for real-time data exploration, on-line prediction, and adaptive knowledge extraction (Jain and Tata, 2017; Ferreira and Ruano, 2009). This refers to *query-driven* predictive analytics (Anagnostopoulos and Triantafillou, 2017a, 2017b), which has been adopted in large-scale distributed computing systems.

However, major challenges arise adopting this *baseline* approach for supporting query-driven & real-time inferential analytics. Firstly, massive raw data transfer is needed for building and updating such central

models. Since this is prohibitive for Internet of Things, i.e., energy-/bandwidth-constrained environments due to constraints like limited network bandwidth, computational power, latency and energy, *Edge Computing* (EC) comes into play (Garcia Lopez et al., 2015; Anand et al., 2017; Gianget al, 2015). Such paradigm can be adopted to cope with this challenge by *pushing* as much intelligent computing logic for inferential analytics as possible close to computing & sensing Edge Devices (EDs) and/or Edge Gateways (EGs) (Renart et al., 2017; Weiet al., 2017), that is to the *network edge* as illustrated in Fig. 1. It is desirable then for the EDs to deliver *only* data summaries, e.g., sufficient statistics & regression model coefficients to the Cloud for query-driven inferential analytics. Moreover, current inferential analytics methodologies like statistical summaries (synopses, multidimensional histograms, topographical maps (Anagnostopoulos et al., 2018), data digest methods), data-driven sampling schemes, and query-driven methods (Savva et al., 2020; Savva et al., 2019) use global models built over *all* transmitted data, as will be elaborated in the related work section. This limits the perspective of local data/knowledge *diversity* experienced on each ED or EG, reflecting the local context awareness of ED's or EG's surroundings. Disregarding the inherent diversity of local models due to the (geo-)distribution of EDs and EGs degrades the locality of contextual information sensed/captured thus eliminating the specificity of local inferred knowledge, which is of high importance in model selection and quality of inferential analytics as will be evidenced in this paper. Even if we desire to exploit such diversity by building, e.g., dif-

E-mail address: christos.anagnostopoulos@glasgow.ac.uk.<https://doi.org/10.1016/j.jnca.2020.102696>

Received 30 September 2019; Received in revised form 12 March 2020; Accepted 3 May 2020

Available online XXX

1084-8045/© 2020 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

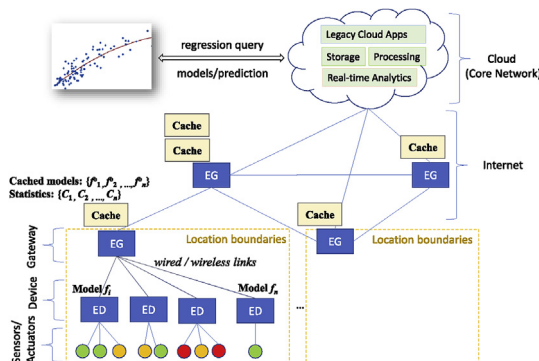


Fig. 1. Figure adopted by [Harth and Anagnostopoulos \(2018\)](#); Physical world is divided in geographical units where sensing/computing devices are deployed. Edge-centric analytics are supported by Flow Tasks 1 and 2 involving Edge Gateways, Edge Devices/Sensors/Actuators delivering models & sufficient statistics.

ferent models, data should be firstly transmitted to the Cloud and then processed and maintained centrally.

1.1. Motivations & goals

We envisage an edge-centric inferential analytics paradigm, where cliques of EDs and EGs are employed as first-class analytics platforms (Harth and Anagnostopoulos, 2017; Sharma and Wang, 2017). Our motivation is based on establishing a methodology that supports inferential analytics materialized at the network edge including e.g., physical sensors (sensing contextual information), mobile EDs, unmanned vehicles for participatory sensing, and Edge Gateways (EGs) interacting with EDs; see Fig. 1.

Moving real-time data from EDs and EGs to remote data centers incurs network latency, which is undesirable for interactive, real-time data exploration and inferential analytics applications; e.g., urban surveillance applications generate humongous volumes of data (speed cameras; environmental time-series; earthquake monitoring) that are bandwidth prohibitive to completely move to the Cloud in real-time (Sharma and Wang, 2017). The network connectivity is intermittent causing loss of functionality if Cloud connectivity gets lost.

Cloud should not be the panacea of inferential analytics paradigm shift. We advocate edge-centric inference and data analysis by pushing the analytics frontiers from centralized nodes to the network periphery, fostering at the same time the inherent models diversity at the network edge. The *pushed* inferential intelligence is distributed among EDs and EGs in order to (i) provide services to regression & exploratory queries issued by analysts/applications and (ii) advocate model/knowledge diversity in a distributed computing ecosystem, which will be adopted for appropriate models selection. This triggers the idea that EDs, or a clique of co-operating EDs, locally build on-line predictive models, which are *maintained* and *selectively* delivered to the EGs for efficient *model selection and sophisticated aggregation*, instead of sending raw data from EDs to EGs and/or to Cloud. Based on this diversified knowledge-only communication between EDs and EGs, we desire to obtain the same quality of analytics, e.g., prediction accuracy and model/curve fitting, compared to the centralized approach by being communication efficient.

We stress that our edge-centric approach retains the core advantages of using Cloud as a support infrastructure but puts back the inferential analytics processing to the edge given that computing capacity of EDs and EGs still increases (Sharma and Wang, 2017). Our approach establishes two fundamental flow tasks (and their sub-tasks) to support edge-centric analytics:

- **Task Flow 1:** Knowledge Transfer from EDs to EGs comprises (1) incremental local model building at the EDs, (2) communication efficient models updating, reacting timely to incoming information, thus preventing concentration of raw data to central locations, and (3) respecting privacy of sensitive information generated/gathered at EDs.
- **Task Flow 2:** Query-driven Analytics Provision at EGs, which are equipped with novel model selection strategies over diverse models to determine the most appropriate models received from EDs to be engaged per query issued by analysts/applications during data analyses.

Task 1 refers to *data thinning* by determining which is the statistically sufficient knowledge to transfer from the EDs to EGs and *when* to update such pieces of knowledge in a dynamic environment. Task 2 refers to provisioning of inferential query-driven analytics at the edge over streams of queries deciding on the most appropriate collection of models cached at the EG per regression query. Both task flows converge at the EGs materializing edge-centric analytics.

The predictive analytics and inferential challenges at the edge infrastructure in real setups are associated with the capability of the edge computing environment to extract the most *relevant* data for model training and inference without significant delays so as to not break the analysts-EGs interactivity constraint, which in real setups is set around 500 ms (Liu and Heer, 2014). This constraint supports that any answers returned over that limit can have negative effects on analysts experience, productivity and decision making. Concretely, analysts engage in *exploratory analysis* (Idreos et al., 2015) to better understand the data. Such analysis is an invariable step in the process of further constructing hypotheses or constructing and training inferential models to answer business questions. As an indicative real-life scenario, we consider predictive queries received over EGs regarding crime-index indicators in regions in the city of Chicago.¹ A workload for this data set² consists of range-queries with aggregation functions over spatial coordinates (Savva et al., 2018). Local models are trained to forecast crime indicators over city regions while dedicated EGs aggregate these models to provide holistic insight on the crime trend in larger areas of the city (or even the whole city). The analysts-EGs interaction to obtain these trends is achieved by query workloads over EGs, which guide the inferential analytics process ranging from which ED to gather which data (Task Flow 1) to which local models to be aggregated to secure timely predictive analytics that ‘follow’ the analysts’ exploratory methodology (Task Flow 2). Within this spectrum, the challenges elicited from the data-management perspective include (among others): missing value imputation, fast and efficient trained models update and adaptation; efficient features selection and data dimensionality reduction, identification of relevant data, and prediction of the induced query workload over the EGs. Our proposed scheme supports with two task flows this challenges spectrum.

2. Challenges & problem fundamentals

2.1. Challenges & desiderata

Multidimensional contextual data have special features such as *bursty nature* and *statistical transiency*, i.e., values expire in short time while statistical dependencies among attributes change over time (Kaneda and Mineno, 2016; Cormode, 2013; Lazerson et al., 2016). Hence, the challenges for edge-centric inferential analytics are: (i) local model learning on EDs requiring real-time model updating and selective model forwarding to EGs in light of minimizing communication overhead (Task Flow 1 challenges), (ii) *best* diverse models selection at EGs

¹ <https://www.neighborhoodscout.com/il/chicago/crime>.

² <http://archive.ics.uci.edu/ml/datasets/Query+Analytics+Workloads+Dataset>.

per regression query, and (iii) model caching techniques that achieve as high analytics quality/accuracy as the centralized approach (Task Flow 2 challenges).

The desiderata of our approach are: (1) a model update mechanism from EDs to EGs and model caching at EGs proved to significantly reduce the communication overhead as only model's parameters and sufficient statistics are disseminated instead of raw data. This meets the desired latency and energy efficiency, and reduces the closed-loop latency to analyze contextual data in real-time. (2) Model selection at EGs allows for fusing diverse local models per query w.r.t. sufficient statistics coming from EDs, thus, retaining the locality of knowledge on each ED without transferring and processing data at EGs.

2.2. Rationale & problem fundamentals

Consider inferential analytics, e.g., (Ferreira and Ruano, 2009; Wang and Li, 2016; Kaneda and Mineno, 2016) in a $(d + 1)$ -dimensional data space $(\mathbf{x}, y) \in \mathbb{R}^{d+1}$, where the analysts/applications seek to learn the dependency between input $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$ and output $y \in \mathcal{Y} \subset \mathbb{R}$ estimated by the *unknown* global data function $y = f(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^d$. For instance, input $\mathbf{x} = [x_1, x_2]^T \in \mathbb{R}^2$ can refer e.g., to attributes temperature x_1 and CO₂ emission x_2 , while y is humidity, or $\mathbf{x} = [x_{t-1}, x_{t-2}, x_{t-3}]^T \in \mathbb{R}^3$ can refer to SO₂ concentration at the previous time instances and $y = x_t$ the current value.

Let us consider regression/exploratory queries issued by analysts/applications over real-time contextual data. Such a query can be represented via a point $\mathbf{q} \in \mathcal{X} \subset \mathbb{R}^d$ such that we locally desire to explore the behavior of $f(\mathbf{x})$ around \mathbf{q} and are provided the prediction $\hat{y} = f(\mathbf{q})$ with prediction error $e(\mathbf{q}) = y - f(\mathbf{q})$; e.g., predict humidity y given $\mathbf{q} = [q_1, q_2]^T$; temperature q_1 and CO₂ q_2 . Moreover, an exploratory query in terms of forecasting or diagnosis is also represented as the time-dependent point $\mathbf{q} = [x_{t-1}, x_{t-2}, \dots, x_{t-d}]^T \in \mathbb{R}^d$, where we investigate the past/future values $y_t = f(\mathbf{q})$ of a bunch of time series for the recent/future time horizon of d (embedding dimension); for instance, forecast the sulphur dioxide SO₂ in the next hour in a specific city area given its current values and the recent CO values obtaining a forecasting error $e(\mathbf{q})$. Furthermore, let a bunch of query points $\{\mathbf{q}_i\}_{i=1}^L \in \mathcal{X}$ in the input data space $\mathcal{X} \subset \mathbb{R}^d$. Analysts/applications are interested in obtaining an estimation of the underlying function f (function estimation task) such that it best fits the input-output space $\mathcal{X} \times \mathcal{Y}$ for those actual inputs (observed input data points) \mathbf{x} closest to query points \mathbf{q}_i , where f fits (explains) the observed pairs (\mathbf{x}, y) . For instance, estimate the correlation function of temperature and humidity in a specific area on the sea surface as recorded by Unmanned Surface Vehicles (USVs)³ in the last 30 min by obtaining a model fitting deviation, e.g., goodness of fit, constructed from individual $e(\mathbf{q}_i)$ errors.

Such edge-centric inferential analytics learns on-line the unknown local predictive model $y = f_i(\mathbf{x})$ over input-output pairs $\{(\mathbf{x}, y)_i\} \in \mathbb{R}^{d+1}$ measured *locally* at each ED i . However, due to the diverse nature/contextual surroundings of each ED, e.g., environmental urban monitoring sensors in a smart city experience different and/or overlapping data ranges of temperature, CO₂ emission, UV radiation, and humidity in different city regions and sea surfaces (Kaneda and Mineno, 2016), a global model f_G fitting *all* data and interpreting *all* statistical and/or spatiotemporal dependencies among attributes cannot capture the very specific characteristics of data subspaces in each ED i . This raises the necessity of estimating local predictive models f_i per ED i representing their specific local data $\{(\mathbf{x}, y)_i\}$ and knowledge, as will be discussed later.

We ought to efficiently and effectively combine such naturally diverse local models f_i built over *different* data into an EG, thus, the EG being able to interpret the diverse statistical dependencies and provide accurate predictions to queries in real-time. The rationale behind the intelligence on the EDs is that they sophisticatedly decide *when* to deliver their local models f_i to the EG, where EG caches these models, notated as f_i^o , to provide real-time analytics. Evidently, the cached model update mechanism is an imperative task of the ED's intelligence trading-off analytics quality at their EG for communication overhead, especially in dynamic data spaces.

The EG supports then inferential analytics given an *ensemble* of cached local models introducing a sophisticated model selection over n cached local models $\mathcal{F} = \{f_1^o, \dots, f_n^o\}$ delivered by its connected n EDs. The final fused model should perform as accurately as if one were told beforehand which local model(s) from \mathcal{F} was the *best for a specific query* and which was the best global model f_G over all the collected data from all EDs. Obviously, given a query, the best possible subset of local models to be engaged for prediction/forecasting/function estimation cannot be known in advance on the EG and its estimation will be proved to be NP-hard later. Moreover, due to the above-mentioned constraints, we cannot build the global f_G on the EG or even at the Cloud over all the data; the EDs do not transfer raw data for efficiency. As a (naive) alternative, the model selection can be simply averaging all local models: average model $f_{AVG}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n f_i(\mathbf{x})$. However, as will be analyzed in Section 7 and shown in our experiments in Section 8, the f_{AVG} induces unnecessarily large variability in prediction resulting in significantly degraded quality of analytics.

2.3. Problem formulation

Let us focus firstly on the Task Flow 1 in Section 1.1. The rationale behind each ED i intelligence is to locally and incrementally build a model f_i and then to efficiently decide, based on sufficient statistics, *when* to update its connected EG with the up-to-date model. Our first challenge is to provide a time-optimized and communication efficient mechanism for statistics-dependent model update.

Problem 1. Given a local model f_i at the ED i , whose image f_i^o is cached to its EG, define a communication efficient model update & delivery mechanism on ED i to replace the cached model at the EG maximizing the analytics quality.

Consider now the Task Flow 2 in Section 1.1. The rationale behind the EG intelligence is to selectively engage some of the cached local models received from its connected EDs given a query by appropriate weighting than averaging while the case of global f_G modeling over all data in the EG is not feasible; no data are transferred from the EDs to EG. Given a query \mathbf{q} , our second challenge is to *predict* the most appropriate local models subset $\mathcal{F}' \subseteq \mathcal{F}$ at EG to engage by being as accurate compared to f_G as possible given (i) communication constraints, (ii) cached model replacement, and (iii) without knowing the distribution of the queries over input data $\{\mathcal{X}_i\}_{i=1}^n$. We desire to predict the most appropriate \mathcal{F}' per query that achieves almost the same or better accuracy than f_G and f_{AVG} without having to send all data from the EDs to the EG.

Problem 2. Given an ensemble $\mathcal{F} = \{f_1^o, \dots, f_n^o\}$ of cached local models on EG, seek a model selection scheme to approximate the best $\mathcal{F}' \subseteq \mathcal{F}$ being as accurate as the global f_G had been built over all collected data by disseminating only local models w.r.t. the update mechanism in Problem 1.

Our third challenge is to establish the sufficient statistics that continuously represent knowledge of performance of the local models, which will be used for the ED's model update mechanism (Problem 1) and for the EG's model selection mechanism (Problem 2). Specifically, each ED i seeks to derive sufficient knowledge of the underlying data and the associated trained local model f_i in order to optimally assess *when* the model should be updated. Such sufficient derived knowledge should be

³ In Section 8, we experiment with inferential analytics over data captured by USVs funded by the EU/GNFUV project.

also exploited by the EG in order to judge *whether* the up-to-date cached model f_i^o is to be involved in the local model subset \mathcal{F}' given a random predictive analytics query.

Problem 3. Given a local model f_i at ED i , define the sufficient statistics and derived knowledge of the underlying data and the associated local model performance, which will be used by the (i) efficient model update mechanism (Problem 1) and the (ii) model selection mechanism in the EG (Problem 2).

3. Related work & contribution

3.1. Related work

In centralized approaches (Kaneda and Mineno, 2016; Bottou and Bousquet, 2007) all collected data are transferred centrally for analysis, thus, centralized predictive modeling and maintenance suffer from heavy burden of massive data transfer and expensive fusion centers including significant delay in providing real-time inferential analytics (Bilal et al., 2018). In some cases, network nodes might not be willing/are not allowed to share their original data due to privacy issues though (Bilal et al., 2018). Our approach pushes analytics to the edge coping with such constraints via communication efficient methods for model updates and high quality of analytics via diversity-oriented model selection.

Distributed approaches for predictive analytics (Wang and Li, 2016; Gabel et al., 2015; Lazerson et al., 2016) focus explicitly on the distributed estimation of a specific global model's parameters over nodes, where the goal is to achieve the same prediction performance as the corresponding centralized one given that gathering all data centrally is expensive and/or impossible. Distributed predictive modeling (1) does not exploit data subspace locality and local models diversity (which are the key components in ensemble-based inferential analytics as evidenced in our experiments), (2) focuses on training a pre-defined global model, where all involved nodes have to agree *in advance* thus there is no option/allowance for heterogeneity in predictive models per EDs, and (3) requires extra techniques for parameters update and synchronization protocols especially in real-time inferential analytics. Such approach enforces nodes to adopt the same predictive algorithm, which is not required in our approach providing the flexibility of hiring different predictive models in EDs; our approach relies on the prediction performance of local models independently of the adopted predictive algorithms on EDs. Moreover, semi-distributed approaches like federated learning (Konečný et al., 2016) involve the building of a global model centrally, which is incrementally updated through partially built local models in EDs. In this context, all nodes focus on maintaining such a model by regular updates, which is in turn disseminated back to the EDs. Apart from the inherent limitation of focusing on a commonly agreed unique model and the lack of communication efficient mechanisms for model update, such approach disregards data locality and models diversity at the EDs. In addition, it does not support model selection apart from the naive model averaging, where the poor quality of analytics (Mendes-Moreira et al., 2012) is showed in Section 8. Such limitations are not present in our methodology as evidenced in our comparative assessment.

Recently, approaches for pushing analytics to the edge are proposed (Kamath et al., 2016) either reduced to distributed predictive modeling (Wang and Li, 2016) (whose limitations are discussed above) or to selective data forwarding (Harth and Anagnostopoulos, 2017; Raza et al., 2015; Harth et al., 2017). Specifically, Harth and Anagnostopoulos (2017) deals with time-optimized data forwarding among EDs and EGs in light of maximizing the quality of inferential analytics. Such approach reduces data communication, however, data processing and model training are still built on EGs. This requires careful data transfer to control model maintenance & adaptation. Our work further pushes model building, sophisticated model update and maintenance to the network periphery (EDs) thus avoiding completely data transfer (cop-

ing also with data privacy), while only parameters & sufficient statistics are *conditionally* disseminated for models adaptation and selection via time-optimized communication efficient mechanisms. The methods in Raza et al. (2015) and Harth and Anagnostopoulos (2017) deal with data suppression based on local forecasting models on sensors in light of re-constructing data at the sink. However, they do not focus on inferential analytics and statistical dependencies learning at EDs (sensors) but only on reducing data communication via data suppression using forecasting models, also adopted in Harth et al. (2017). These models selectively disseminate data and univariate re-construction models used at the sink, thus, actual predictive modeling is achieved at the EG/sink with no guarantee on the analytics quality/prediction performance. Moreover, predictive modeling does not scale since the EG lacks of model selection and caching mechanisms for selecting and maintaining the best models per query, other than simple model averaging, whose limitations were discussed above, in Brown et al. (2005), and shown in Section 8.

This work, as the first edge-centric, real-time, and communication efficient inferential analytics methodology, significantly extends our previous work in Harth and Anagnostopoulos (2018). The fundamental extensions are: In Task Flow 1, (1) we depart from the instantaneous model update mechanism in Harth and Anagnostopoulos (2018) by introducing an error-tolerance model update mechanism based on the theory of optimal stopping (Shiryaev, 2008), which evidences significantly higher communication efficiency and higher quality of inferential analytics compared with (Harth and Anagnostopoulos, 2018); (2) we provide the optimality achieved by our mechanism via mathematical analyses; (3) we introduce a mechanism that provides immediate feedback to the novelty/familiarity technique at EDs in light of reducing the misjudgments for local model adaptation. In Task Flow 2, (4) we establish a diverse model selection theory at the EG per query and prove that such selection is NP-hard, and (5) we introduce computationally efficient error-aware model selection schemes; (6) we analyze the expected communication overhead in Harth and Anagnostopoulos (2018) and in this work, and (7) provide analytical upper bounds in sufficient statistics updates.

It is worth mentioning that the proposed synergies among EDs and EGs trigger the introduction of *incentives* in EC. Incentivisation mechanisms in such environment introduced in Liu et al., n.d. could encourage EDs to participate in e.g., knowledge, crowd-sensing (Liu et al., n.d.) trading off privacy, edge analytics provision and predictive analytics quality. Moreover, our scheme could incorporate incentive mechanisms for content sharing, e.g., video Ads, in opportunistic device-to-device networks as introduced in Liu et al. (2018), which integrates users' mobility with crowd-sourcing. Finally, synergies among networks of EDs and EGs, e.g., femtocell and macrocell edge nodes in mobile computing environments require optimized strategies to allocate transmission powers efficiently and share network resources as proposed in Liu et al. (2019b); this will guarantee optimized network deployment and resilient provision of inferential analytics.

3.2. Contribution

Our eminent contribution is:

- An optimal communication-efficiency aware scheme based on the theory of optimal stopping for model updates solving Problem 1;
- Model selection algorithms at EGs solving Problem 2;
- A novel input-error hetero-associative statistical learning algorithm and its convergence analysis extracting sufficient statistics solving Problem 3;
- Stochastic algorithms that establish the optimality of our schemes w.r.t. communication efficiency and analytics quality;
- A feedback-based mechanism for tuning the model adaptation at the EDs;

- Analysis of the expected communication overhead and estimation of the upper bound of the expected statistics;
- Comprehensive comparative assessment against methods: global/baseline, model averaging (Konečný et al., 2016; Harth and Anagnostopoulos, 2018; Harth and Anagnostopoulos, 2017) and (Raza et al., 2015) using three real datasets derived from static and mobile EDs/computing/sensing devices and unmanned vehicles. We experiment with well-known regression and time-series forecasting models over various data dimensionality.

4. Predictive intelligence at the edge device

4.1. Predictive familiarity & model update: overview

In this section, we provide a bird's eye view of our approach to edge-centric inferential analytics introducing the model update mechanism at the EDs (Task 1) and the model selection mechanism at the EGs (Task 2). The ED i in Fig. 1 locally learns a parametric model $f_i(\mathbf{x}; \mathbf{b}_i)$ based on the recent local data in a sliding window $\mathcal{N}_i = \{(\mathbf{x}, y)_{t-N+1}, \dots, (\mathbf{x}, y)_t\}$ with the most recent N observed input-output pairs (\mathbf{x}, y) . Let us denote $\mathbf{b}_i \in \mathcal{B}$ the parameters of the current local model f_i belonging to a parameter space and \mathbf{b}_i^o the parameters of the cached local model f_i^o where ED i has already sent to EG at some time in the past. For instance, in the case of linear regression $f_i(\mathbf{x}) = \mathbf{b}_i^\top \mathbf{x}$ with parameter $\mathbf{b}_i \in \mathcal{B} \subset \mathbb{R}^d$. The ED i is responsible for updating the EG when there is a significant discrepancy of the prediction performance of the local f_i and cached f_i^o at EG. The ED i keeps a copy of f_i^o locally to drive its decision making discussed later and sends the parameters \mathbf{b}_i and some sufficient statistics, if it is deemed necessary, *only* to its EG. This decision has to be taken in real-time by sequentially observing input-output pairs and the current prediction discrepancy between the local and cached models.

Consider a discrete time domain $t \in \mathbb{T} = \{1, 2, \dots\}$. The ED i at time t captures the t th input-output pair $(\mathbf{x}, y)_t$ and, in real-time:

- **Case A:** Decides whether the pair $(\mathbf{x}, y)_t$ significantly changes the prediction performance of the current local f_i or not. In this case (A.I), the ED i appends $(\mathbf{x}, y)_t$ to window \mathcal{N}_i discarding the oldest pair and incrementally adjusts or partially re-trains f_i accordingly based on the updated \mathcal{N}_i . Otherwise, (A.II), f_i is not adjusted or re-trained given $(\mathbf{x}, y)_t$.
- **Case B:** Decides whether the updated local f_i (decided in the case A.I) should be sent to EG or not. In this case (B.I), ED i updates EG with the up-to-date f_i provided that a significant prediction performance discrepancy is observed compared with the cached f_i^o . Otherwise, (B.II) no model update and no delivery is performed between ED i and EG.

In Case A, the ED i should be able to instantaneously determine whether the new pair is drawn from the input-output subspace $(\mathcal{X}_i, \mathcal{Y}_i)$ defined by the pairs in \mathcal{N}_i or not. In the former case, the new pair *interpolates* within the current input-output data subspace thus being considered as *familiar*. This familiarity indicates that the current model f_i is expected to provide a good prediction $\hat{y}_t = f_i(\mathbf{x}_t)$ given the t th input \mathbf{x}_t , i.e., $|y_t - \hat{y}_t| \leq \rho_0$ for some arbitrary accuracy threshold $\rho_0 > 0$. In this case, ED i does not need to adapt or re-train the current model f_i given that the t th pair is familiar (Case A.II), thus no communication with EG is needed.

If the t th pair is considered unfamiliar or *novelty* w.r.t. the current input-output subspace, it renders a re-training or adaptation of the current model f_i (case A.I) (depending on the regression model f_i). For instance, f_i is adapted to new pairs using recursive least squares & incremental support vectors (Kaneda and Mineno, 2016; Engel et al., 2004), incremental/gradient Radial Basis Function (Schwenker et al., 2001), or re-training is required over \mathcal{N}_i ; see Appendix B. In general, a new local model f_i is derived after adaptation or re-training, thus, yielding ED i to examine: (1) the instantaneous model performance discrepancy

between the new f_i and the cached model f_i^o (Case B) and (2) the past behavior of such discrepancy to obtain a holistic insight on whether to update the EG or not with the updated local model f_i . We quantify this discrepancy as the absolute difference of errors of f_i and f_i^o :

$$z_t = |e_i(\mathbf{x}_t) - e_i^o(\mathbf{x}_t)|. \quad (1)$$

Based on the current discrepancy z_t and its evolution $\{z_1, z_2, \dots, z_t\}$ since the last model update, the ED i decides on updating EG with the new model f_i and locally updating the cached model $f_i^o = f_i$ (Case B.I). Otherwise, there is no need ED i to update EG, even if the cached and new models do behave *similar* regarding the prediction performance expressed by this discrepancy. We enforce ED and EG to both have inferential and predictive models that behave the same in terms of prediction performance for the same input.

Remark 1. Our approach is generic in predictive models. Our algorithms extract knowledge only from the input space and prediction error being independent on the nature of the predictive algorithms/models/parameters on the EDs and their statistical expressiveness, which is application-specific/data-analysts decision on which models to adopt for inferential analytics. This supports the flexibility and heterogeneity of edge-centric inferential analytics and modeling.

Focusing on Task 1, ED goes with: assessing the familiarity of incoming input-output pairs (Section 4.2) and deciding on model updates (Section 5).

4.2. Familiarity Inference & local sufficient statistics

The first challenge is to define an on-line method for assessing the novelty of a new pair, i.e., implementing the decision in Case A. Based on the outcome decision of Case A, the ED i might trigger a model update to EG. The novelty of an incoming pair (\mathbf{x}, y) might trigger both: local model adaption and cached model update. In order to assess the novelty of a pair in terms of prediction accuracy, our idea is to associate the input vector space $\mathbf{x} \in \mathcal{X}$ with the prediction error space $e_i(\mathbf{x}) \in \mathbb{R}$ w.r.t. model f_i , thus, being capable of approximating the expected prediction error given an unseen input \mathbf{x} . We learn this association by *jointly* quantizing the input-error space generating input and error representative of the $\mathcal{X} \times \mathbb{R}$ space. Specifically, the ED i *incrementally* learns the k -th vector input subspace and *simultaneously* associates the model prediction error with that input subspace. To achieve this (hetero) association, we need to on-line quantize the input space into K unknown subspaces, each one represented by an *input prototype* $\mathbf{w}_k \in \mathbb{R}^d, k \in [K]^4$ and then associate the prediction error $e(\mathbf{x}) = y - f_i(\mathbf{x})$ over input \mathbf{x} lying around prototype \mathbf{w}_k with an *error prototype* $u_k \in \mathbb{R}$. That is, a new input \mathbf{x} is firstly mapped to the closest \mathbf{w}_k and then the corresponding error $e(\mathbf{x}) = y - f_i(\mathbf{x}) : k = \arg\min_{k \in [K]} \|\mathbf{x} - \mathbf{w}_k\|$ obtained by the model with this input is summarized by u_k .

Example: Fig. 2 (left) shows the time series $y = x_t = f_i(x_{t-1}, x_{t-2})$ with embedding delay dimension $d = 3$ and input $\mathbf{x} = [x_{t-1}, x_{t-2}]^\top \in \mathbb{R}^2$ (the time series segment is extracted from the real dataset D3 in Section 8.1) along with the input space prototypes $\mathbf{w}_k = [x_{t-1,k}, x_{t-2,k}]^\top \in \mathbb{R}^2$. The ED locally learns the autoregressive-recursive least squares (AR-RLS) model; see Section 8.1. Fig. 2 (right) shows the input-error space $\mathcal{X} \times \mathbb{R}$ where for a specific area around the input prototypes \mathbf{w}_k , we estimate the associated error plane u_k . The outcome of this associative quantization is that we estimate the expected prediction error $\mathbb{E}_x[e(\mathbf{x})]$ of any model given an unseen input \mathbf{x} . The error prototype u_k refers to the conditional expectation $\mathbb{E}_x[e(\mathbf{x})|\mathbf{w}_k] = \arg\min_{l \in [K]} \|\mathbf{x} - \mathbf{w}_l\|$ of the model error around the input space represented via \mathbf{w}_k , which is the closest prototype to input \mathbf{x} . As shown in Fig. 2 (right), there are subspaces of \mathcal{X} where the model prediction performance is relatively satisfactory and in some areas where the expected

⁴ $k \in [K]$ is a compacted notation for $k = 1, \dots, K$.

error is relatively high. Such statistical information not only drives the input familiarity/novelty inference at the ED but also provides the basis for model selection at the EG.

4.3. Hetero-associative input-error learning

We associate the *local* performance of f_i in the input subspace, represented by $\mathbf{w}_k \in \mathbb{R}^d$, with the *local* prediction error, represented by $u_k \in \mathbb{R}$; We propose a novel methodology for incremental (hetero) associative input-error space quantization at ED i with unknown number of prototypes K . The objective joint optimization function in our case minimizes the combined (i) conditional Expected Quantization Error (EQE) in the input space, used for learning the best input prototypes representing novelty in input space, and (ii) conditional Expected Prediction Error (EPE) used for learning the best error prototypes capturing local model performance. The condition is based on the closest input prototype, i.e., we optimize the input/error prototypes, which are hereinafter referred to as *sufficient statistics*:

$$C_i = \mathcal{W}_i \cup \mathcal{U}_i, \quad (2)$$

with $\mathcal{W}_i = \{\mathbf{w}_k\}$ and $\mathcal{U}_i = \{u_k\}$ minimize the joint EQE/EPE:

$$\mathcal{J}(\{\mathbf{w}_k, u_k\}) = \mathbb{E} \left[\frac{\lambda}{d+1} \|\mathbf{x} - \mathbf{w}_k\|^2 + \frac{(1-\lambda)d}{d+1} |e(\mathbf{x}) - u_k| \mathcal{A}_k \right]. \quad (3)$$

In (3), the condition $\mathcal{A}_k \equiv \{k = \arg \min_{l \in [K]} \|\mathbf{x} - \mathbf{w}_l\|^2\}$, while $e(\mathbf{x}) = y - f_i(\mathbf{x})$ is the prediction error, and $\lambda \in [0, 1]$ is a regularization factor for weighting the importance of the input-error space quantization. Notably, $\lambda = 1$ refers to the known EQE (Shen and Hasegawa, 2006), while $\lambda \rightarrow 0$ indicates pure prediction-error based quantization. The expectation in (3) is taken over input-error pairs $(\mathbf{x}, e(\mathbf{x})) \in \mathbb{R}^d \times \mathbb{R}$ and the multiplied fractions $\frac{1}{d+1}$ and $\frac{d}{d+1}$ are present for normalization.

Remark 2. The input prototypes $\{\mathbf{w}_k\}$ in (2) do not solely refer to the optimal representatives of input space as, e.g., they could have been derived from K -means (Shen and Hasegawa, 2006), Self-organized Maps (Kohonen et al., 2001) or Adaptive Resonance Theory (Carpenter and Grossberg, 1988). Instead, based on (3), the position of input prototypes in \mathcal{X} is optimal that minimizes **both**: quantization error and prediction error. It is expected to observe a high density of input prototypes in the input space where the model prediction performance is poor compared to other subspaces, where the model behaves more accurately. This is reflected by the joint optimization objective that drags the input prototypes in areas where the model accuracy varies significantly.

Obviously, the number of prototypes K is not known a-priori and the ED i incrementally decides *when* to add a new input-error prototype based on the input novelty and model performance. Hence, we propose an evolving algorithm that minimizes (3) starting initially with one ($K = 1$) input/error prototype pair (\mathbf{w}_1, u_1) corresponding to the first input \mathbf{x}_1 and prediction error $u_1 = f_i(\mathbf{x}_1) - y_1$ given the first pair (\mathbf{x}_1, y_1) . Then, current prototypes and new ones are conditionally adapted and created, respectively, w.r.t. incoming pairs materializing the concept of *familiarity* and *novelty*, respectively. Specifically, based on a familiarity threshold ρ_I between the new input \mathbf{x} and its closest prototype \mathbf{w}_k and a dynamically changing error tolerance ρ_O for the current error $y - f_i(\mathbf{x})$, the pair (\mathbf{x}, y) is classified as novel or not with the so far observed pairs. If the new pair is considered familiar w.r.t. recent history, the closest input prototype and corresponding error prototype are adapted to the familiar pair. However, if the current prediction error over the closest input subspace is not tolerated, i.e., greater than ρ_O , then this tolerance ρ_O decreases denoting less tolerance in the error space for future inputs. On the other hand, if input \mathbf{x} is relatively far from its closest \mathbf{w}_k w.r.t. ρ_I then a new input-error prototype is created. If the current prediction error is not tolerated, i.e., greater than ρ_O , then this pair is considered novel, which immediately renders the model re-learning/adaptation. Otherwise, this pair is familiar since the

current error is tolerated, thus, avoiding model adaptation/re-training. Nonetheless, ρ_O decreases denoting less tolerance in the error space for future novel inputs.

Familiarity ρ_I represents a threshold of similarity between input \mathbf{x} and prototype \mathbf{w}_k , thus, guiding us in determining when a new input-error prototype pair should be formed. Then, combined with the prediction error tolerance, the methodology decides on a novel or familiar input w.r.t. the prediction performance of the local model. Moreover, the gradual decrease of ρ_O upon deciding familiarity/novelty of the input-output pair or model adaptation, signals the decrease in prediction error tolerance to enforce model adaptation with higher probability in future pair observations. This avoids monopolizing familiarity decisions thus urging model adaptations and possible updates maintaining high quality of inferential analytics at the EGs.

4.4. Familiarity & novelty inference algorithm

The evolving Algorithm 1 minimizes the objective (3) by incrementally adapting the input and error prototypes as stated in Theorem 1. Note, \mathbf{w}_k and u_k converge to the centroid (mean vector) of the inputs \mathbf{x} and to the median of the absolute prediction error in the k -th input-error subspace, respectively, as stated in Theorem 2. These (converged) prototypes are the sufficient statistics C_i (Problem 3), which will be exploited by the EG for determining the most appropriate diverse models given a query.

Algorithm 1 Hetero-associative Familiarity Inference.

Input: new pair (\mathbf{x}, y) , familiarization thr. ρ_I , error tolerance thr. ρ_O , minimum error tolerance ρ_O^*
Output: familiarity; updated prototypes C_i

- 1: familiarity \leftarrow FALSE
- 2: closest input prototype $k = \arg \min_{\ell \in [K]} \|\mathbf{x} - \mathbf{w}_\ell\|$
- 3: model prediction: $\hat{y} = f_i(\mathbf{x})$; absolute error $e = |y - \hat{y}|$
- 4: if $(\|\mathbf{x} - \mathbf{w}_k\| \leq \rho_I)$ then
- 5: if $e > \rho_O$ then
- 6: $\rho_O = \max(\frac{1}{2}\rho_O, \rho_O^*)$; adapt model f_i w.r.t. (\mathbf{x}, y)
- 7: else
- 8: prototypes adaptation in (4); familiarity \leftarrow TRUE
- 9: end if
- 10: else
- 11: novelty (new prototype): $K = K + 1$, $\mathbf{w}_K = \mathbf{x}$, $e_K = e$
- 12: if $e \leq \rho_O$ then
- 13: $\rho_O = \max(\frac{1}{2}\rho_O, \rho_O^*)$; familiarity \leftarrow TRUE
- 14: else
- 15: adapt model f_i w.r.t. (\mathbf{x}, y)
- 16: end if
- 17: end if

Theorem 1. The prototypes $(\mathbf{w}_k, u_k) \in C_i$ minimize (3) iff given a pair (\mathbf{x}_t, y_t) they are updated as:

$$\Delta \mathbf{w}_k = \alpha_t \frac{\lambda}{d+1} (\mathbf{x}_t - \mathbf{w}_k), \Delta u_k = \alpha_t \frac{(1-\lambda)d}{d+1} \text{sgn}(e_t - u_k), \quad (4)$$

$\alpha_t \in (0, 1)$ is a learning rate: $\sum_{t=1}^{\infty} \alpha_t = \infty$ and $\sum_{t=1}^{\infty} \alpha_t^2 < \infty$, $e_t = y_t - f_i(\mathbf{x}_t)$, and $\text{sgn}(\cdot)$ is the signum function.

Proof. See Appendix A.1 □

Theorem 2. (Convergence). The prototypes $(\mathbf{w}_k, u_k) \in C_i$ converge to the centroid of input vectors and median of prediction error, respectively, of the k -th input-error subspace.

Proof. See Appendix A.2 □

The Algorithm 1 on ED i (i) optimally quantizes the input-error space by minimizing (3), (ii) on-line decides whether (\mathbf{x}, y) is familiar or not used for triggering model adaptation/re-training and/or cached model

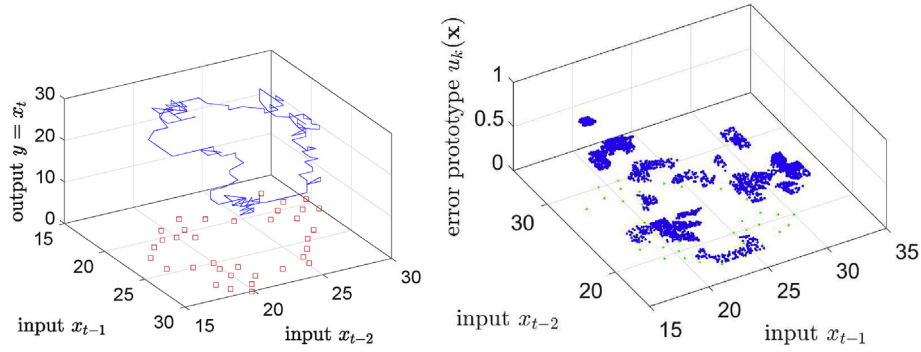


Fig. 2. (Left) Segment of time series $x_t = f(x_{t-1}, x_{t-2})$ and input space prototypes $w_k = [x_{t-1,k}, x_{t-2,k}]$; (right) input-error association with input space prototypes w_k in green dots and quantized error space generated by prototypes $u_k(x)$ for AR-RLS model. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

update, and (iii) incrementally evolves by identifying new prototypes in C_i . It returns the updated statistics C_i and a classification of (x, y) as familiar or novelty. Since novelty might trigger a possible model adaptation, the ED i is expected to obtain a new local model and assesses the performance discrepancy with the cached model $z = |e_i(x) - e^o(x)|$ given (x, y) . Based on the evolution of discrepancy values over time, as will be discussed in Section 5, ED i efficiently decides on delivering the new mode to EG updating its cache. The ED i has now all the available knowledge for its input-error space encoded in C_i .

Remark 3. The advantages of the Algorithm 1 are threefold. Firstly, it incrementally minimizes both the EQE and EPE based on the predictability performance and the distribution of the input and error spaces concurrently, which aligns with the principles of hetero-associative learning. Secondly, it classifies online the familiarity of an input-output pair conditioned to the predictability of the local model, thus, acting as a classifier. Thirdly, it conditionally updates the local model to follow the underlying distribution of the input-output space towards convergence (see Appendix C for an upper bound on the model adaptation rate). Hence, Algorithm 1 adopts three behaviours: (i) as an incremental input-error vector quantizer conditioned on a tolerated prediction error and familiarization decision threshold, (ii) as an online novelty classifier, where it controllably expands the sufficient statistics (prototypes) and, (iii) as an adaptation mechanism, which conditionally updates the local model based on the current predictability model's performance.

5. Time-optimized model update mechanism

5.1. Model discrepancy & tolerance

We introduce a time-optimized mechanism for: (i) deciding *when* to update the cached model f_i^o at the EG with the new updated model f_i at the ED i (and potentially the partial changes in C_i); and (ii) adjusting the Algorithm 1 to reduce novelty mis-classifications via positive and negative feedback.

At time instance t , the ED i is capturing the pair (x_t, y_t) with errors: $e_t(x_t) = y - f_i(x_t)$ and $e_t^o(x_t) = y - f_i^o(x_t)$ obtained from the current model $f_i(x_t)$ and the cached model $f_i^o(x_t)$. Without abuse of notation, we remove the subscript i referring to ED i in the remainder of this section for the sake of readability. The instantaneous discrepancy $z_t = |e(x_t) - e^o(x_t)|$ indicates the difference in the prediction accuracy obtained by f and f^o models given input x_t . The model update mechanism in Harth and Anagnostopoulos (2018) is only based on an instantaneous decision making and specifically on the hard decision: $z_t > \theta$ for a fixed discrepancy threshold $\theta > 0$. That is, the ED updates EG at time instance t iff z_t exceeds θ . The discrepancy threshold θ is application specific and indicates the desired expected error difference between the local model f at ED and the cached model f^o at EG. However, such dif-

ference, which leads to instantaneous decisions for sending the model f (and sufficient statistics C) from ED to EG, might incur significant overhead for updating the cached model f^o in EG. This is happening should the probability $P(z_t > \theta)$ is relatively high in certain periods. Specifically, if we consider n EDs connected to an EG, then the communication load for updating the individual models $\{f_i\}_{i=1}^n$ to the EG does not scale with the number of EDs. Hence, we obtain expected communication $nP(z_t > \theta)$ at any time instance t which cannot be neglected (especially when $P(z_t > \theta)$ is different for each ED in real life scenarios). Moreover, the hard decision threshold θ indicating the absolute difference between f and cached f^o includes the inherent discrepancy variance of both models given a random input x . This is not trivially indiscriminable; the error variances σ_e^2 and $\sigma_{e^o}^2$ are added up when considering the error difference $|e - e^o|$. Any instantaneous excess of z_t over θ does not certainly indicate an experienced change in both models in terms of predictability in the recent past.

Our idea is to *observe the evolution* of the discrepancy values $\{z_t\}$ in multiple time instances between the two models within a (yet, unknown) time horizon. Based on this observation, we expect ED to decide on a model update. This means that we avoid instantaneous and sudden model updates with the major aim to reduce communication overhead due to possibly highly frequent model and sufficient statistics updates to the EG. Evidently, the length of such time horizon cannot be determined a priori due to the stochastic nature of z_t . Let Z_t denote the positive random variable of discrepancy with $z_t > 0$ be a realization value at time instance t . We allow ED to tolerate a cumulated discrepancy:

$$S_t = Z_1 + \dots + Z_t, \quad (5)$$

for a specific time horizon since the last model update (resetting $t = 0$), where the expected discrepancy between the local and cached models is bounded by a *discrepancy tolerance* $\Theta > \theta$. In this context, the ED is given the opportunity to postpone model updates as long as the cumulated discrepancy since the last update S_t is less than Θ . During this time horizon starting from the last model update, the ED saves communication resources by avoiding rapid model updates of sporadically sudden excesses of $Z_t > \theta$. Evidently, this comes at the expense of a potential difference in prediction accuracy of the model f at ED and cached model f^o at the EG. We strictly enforce such expected difference to be bounded by Θ in light of reducing the communication overhead, which might be relatively significant if Z_t stochastically oscillates around θ .

5.2. Model update based on cumulative discrepancy

To observe the evolution of discrepancy variables between the two models, we focus on the cumulative sum S_t up to t , i.e., $S_t = \sum_{\tau=1}^t Z_\tau$ since the last model update, where $t = 1$ indicates the first landmark time instance right after the last model update. S_t is a random variable

made of the summation of random variables $\{Z_\tau\}_{\tau=1}^t$ up to t . Such a sum is adopted as an indicator of the evolution of the discrepancy, which is expected to be bounded by Θ indicating the minimum acceptable quality of inferential analytics based on the cached model at EG. We desire S_t to be as close to Θ as possible to avoid updating the model at EG during this time horizon (thus saving communication resources), but not to exceed this threshold, since there might be a significant predictability discrepancy between the local and cached models.

Evidently, the size, i.e., number of random variables Z_τ , in S_t is increasing with stochastic step and depends on the length of the time horizon ED does not communicate with EG for model update. The value of S_t is governed by the stochasticity of the discrepancy values z_τ in time. Hence, we do not know and cannot forecast when S_t will reach Θ and when S_{t+1} will exceed Θ , i.e., when $S_t \leq \Theta$ and $S_{t+1} > \Theta$ since $S_1 = Z_1, S_2 = Z_1 + Z_2, \dots, S_t = Z_1 + \dots + Z_t$ is a stochastic process independent of Θ with $\mathbb{E}[Z] = \mathbb{E}_x[|e(x) - e^o(x)|] = \mathbb{E}_x[|f(x) - f^o(x)|] < \infty$ given that the expectation of output $\mathbb{E}[y] < \infty$.

Our problem then is to determine the time instance $t^* > 0$ since the last update such that the sum S_{t^*} is as close to Θ as possible but without exceeding Θ . In this context, we cast this problem as a time-based stochastic optimization problem (Bruss and Le Cam, 2000) by finding the *optimal stopping time* t^* , which *minimizes the expected discrepancy between accumulated discrepancy and tolerance Θ without exceeding this boundary*. Obviously, the case $S_t > \Theta$ is undesirable since it induces a *penalty* that we should have stopped earlier before the tolerance had exceeded the boundary. In our mechanism, we *delay* the model update up to the best time instance t^* in hopes of reaching as close to Θ thus reducing the communication overhead. If we gathered more discrepancy than Θ ($S_{t^*} > \Theta$), then we should have stopped before t^* to avoid expected discrepancy greater than Θ between the models.

The benefits of adopting a (stochastic) cumulative discrepancy-based decision making rather than instantaneous decision making as in the models found in the literature (see Section 8.3) is that it enforces ED to either take a decision on model update or to continue with another observation by avoiding redundant model updates. On the other hand, if the application is in need of highly accurate predictions, a certain (controlled) delay must be tolerated tuning the boundary Θ . Obviously, we cannot delay for ever to avoid any communication between ED and EG, since in a dynamic environment the current model f is about to significantly change should the underlying joint distribution of input-output space $(\mathcal{X}_t, \mathcal{Y}_t)$ is changing (indicated with a relatively high number of ‘novelty’ pairs) and both models f and f^o have to represent the current (or better recent past) state of nature.

We naturally provide the function $G_t(S_t)$ in (6) representing the tolerance return at the ED involving the boundary Θ and the penalty when S_t exceeds Θ , thus, penalizing our ‘delayed’ decision for model updates:

$$G_t(S_t) = \begin{cases} S_t = \sum_{\tau=1}^t Z_\tau & \text{if } S_t \leq \Theta, \\ 0 & \text{if } S_t > \Theta. \end{cases} \quad (6)$$

The ED attempts to maximize the expected return $\mathbb{E}[G_t(S_t)]$ by delaying model updates, thus, saving communication resources but not exceeding the established boundary to secure the minimum discrepancy between f and f^o models in ED and EG, respectively. We now formally state the model update problem, which specifies the generic **Problem 1**:

Problem 4. *Given a boundary $\Theta > 0$ and a sequence of discrepancy variables $\{Z_t\}$, find the optimal stopping time t^* such that the supremum of the expected return $\sup_{1 \leq t \leq \infty} \mathbb{E}[G_t(S_t)]$ is attained. The maximum expected return is $\mathbb{E}[G_{t^*}]$.*

When the optimal stopping time t^* is determined, as we will show later in this section, the ED updates the EG with the model f and the mechanism starts-off a new era by observing new discrepancy behavior with resetting the time landmark $t = 1$.

5.3. Solution fundamentals

In order to establish the solution fundamentals for **Problem 4**, we provide preliminaries on the Optimal Stopping Theory (Shiryaev, 2008) to help us classify our **Problem 4** as an optimal stopping problem. In this context, we need to prove first the existence of the optimal stopping time t^* explicitly for our problem and, then, provide our optimal stopping rule, which is the decision rule for updating the model to EG. The reader could skip Section 5.3.1 should they be familiar with the principles of the optimal stopping theory.

5.3.1. Optimal stopping theory

The theory of optimal stopping (Bruss and Le Cam, 2000; Shiryaev, 2008) is concerned with the problem of choosing a time instance to take a certain action in order to maximize an expected return. A stopping rule problem is associated with: (i) a sequence of random variables Z_1, Z_2, \dots , and (ii) a sequence of return functions $(G_t(z_1, \dots, z_t))_{1 \leq t}$, which depend only on the observed values z_1, \dots, z_t of the corresponding random variables. An optimal stopping rule problem is described as follows: We are observing the sequence of $(Z_t)_{1 \leq t}$ and at each time instance t we choose either to stop observing or continue. If we stop observing at time instance t , we gain a return G_t . We desire to choose a stopping rule to maximize our expected return.

Definition 1. *An optimal stopping rule problem is to find the optimal stopping time t^* which maximizes the expected return $\mathbb{E}[G_{t^*}] = \sup_{0 \leq t} \mathbb{E}[G_t]$.*

The available information up to t is the sequence \mathbb{F}_t of the values of the random variables Z_1, \dots, Z_t , a.k.a. filtration.

Definition 2. *The 1-stage look-ahead stopping rule is the stopping criterion*

$$t^* = \inf\{t \geq 0 : G_t \geq \mathbb{E}[G_{t+1} | \mathbb{F}_t]\}. \quad (7)$$

In other words, t^* calls for stopping at the first time instance $t > 0$ for which the return G_t for stopping at t is (at most) as high as the expected return of continuing to the next time instance $t + 1$ and then stopping.

Definition 3. *Let A_t denote the event $\{G_t \geq \mathbb{E}[G_{t+1} | \mathbb{F}_t]\}$. The stopping rule problem is monotone if $A_0 \subset A_1 \subset A_2 \subset \dots$ almost surely (a.s.)*

A monotone stopping rule problem can be described as follows: The set A_t is the set on which the 1-stage look-ahead rule (1-sla) defined in **Definition 3** calls for stopping at t . The condition $A_t \subset A_{t+1}$ means that if the 1-sla rule calls for stopping at t , then it will also call for stopping at $t + 1$ no matter what Z_{t+1} happens to be. Similarly, $A_t \subset A_{t+1} \subset A_{t+2} \subset \dots$ means that if the 1-sla rule calls for stopping at t , then it will call for stopping at all future times no matter what the future observations turn out to be.

Theorem 3. *The 1-sla rule is optimal for monotone stopping rule problems.*

Proof. See (Shiryaev, 2008) □

5.3.2. Optimal stopping rule for model update

Our target is to determine a 1-stage look-ahead optimal stopping rule for model update observing a sequence of discrepancy values. Before proceeding with our optimal stopping rule, we need to prove the existence of the optimal stopping time of **Problem 4**. That is, we check if the ED by applying our proposed 1-sla stopping rule maximizes the expected return being as close to the boundary as possible without exceeding this.

Lemma 1. *The optimal stopping time t^* that maximizes $\mathbb{E}[G_t(S_t)]$ in **Problem 4** exists.*

Proof. See **Appendix A.3** □

Given the existence of the optimal stopping time t^* for **Problem 4**, we provide a solution by defining the optimal stopping rule adopted by

the ED. We report on a 1-sla rule based on the *principle of optimality* (Shiryaev, 2008) in Theorem 4, at which the ED stops observing discrepancy values and then updates the EG at the first time instance t such that: $G_t(S_t) \geq \mathbb{E}[G_{t+1}(S_{t+1})|\mathbb{F}_t]$, with the event $\{S_t \leq \Theta\} \in \mathbb{F}_t$. That is, any additional observation of a discrepancy value at time $t+1$ would not additionally contribute to the maximization of our return in Problem 4. The 1-sla rule is optimal since the stochastic difference $\mathbb{E}[G_{t+1}(S_{t+1})|\mathbb{F}_t] - G_t(S_t)$ is monotonically non-increasing with S_t , as will be proved in Theorem 4. Based on the principle of optimality for our 1-sla stopping rule, we provide the optimal rule in Theorem 4 for model updates:

Theorem 4. Given a sequence of discrepancies Z_1, \dots, Z_t , the optimal stopping rule t^* for Problem 4 is

$$t^* = \inf\{t \geq 0 : S_t \geq \frac{1}{1 - F_Z(\Theta - S_t)} \int_0^{\Theta - S_t} z dF_Z(z)\}, \quad (8)$$

$F_Z(z) = P(Z \leq z)$ is the cumulative probability function of discrepancy Z .

Proof. See Appendix A.4 \square

The optimal stopping rule in Theorem 4 involves the cumulative sum of discrepancies and the conditional expectation of tolerance up to t given S_t . This clearly demonstrates the dynamic (optimal) tolerance threshold departing from the instantaneous one in Harth and Anagnostopoulos (2018), where the ED monitors the stochastic behavior of discrepancies at every time.

The ED decides to update the model at the first time instance the cumulative discrepancy up to t is higher than the expected discrepancy up to t multiplied by a factor $(1 - F_Z(\Theta - S_t))^{-1} > 1$. Since the 1-sla rule in Theorem 4 is optimal by Lemma 1, the ED with fixed Θ guarantees that the expected return is as much close to Θ as possible and *no other stopping rule can guarantee as much*. Based on a Θ , our model update mechanism is flexible to treat and control the expected delay and the expected discrepancy between the local and cached models in ED and EG, respectively. In the remainder of this section, we demonstrate the optimality of model update rule for different discrepancy distributions and the practicality of the mechanism.

5.3.3. Optimal model update rule in action

Let us demonstrate the optimality of our rule in Theorem 4 by adopting different probability distribution functions $F_Z(z)$ of discrepancy Z . Fig. 3 shows the probability distribution (PDF) of discrepancy Z for an ED, where it adopts the models RBF and LM over the dataset D2 (Section 8.1). For illustration purposes, we estimate the parameters for fitting the PDF of Z via a Weibull $W(\alpha_1, \beta_1)$ and Exponential $Exp(\mu^{-1})$ distribution functions, where the corresponding F_Z is then

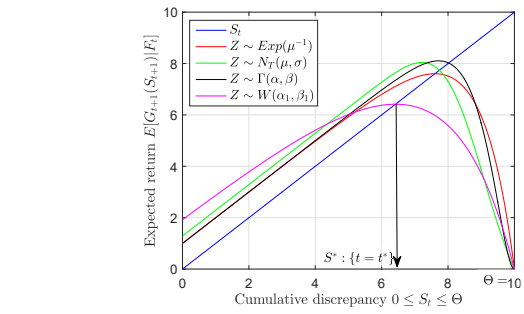
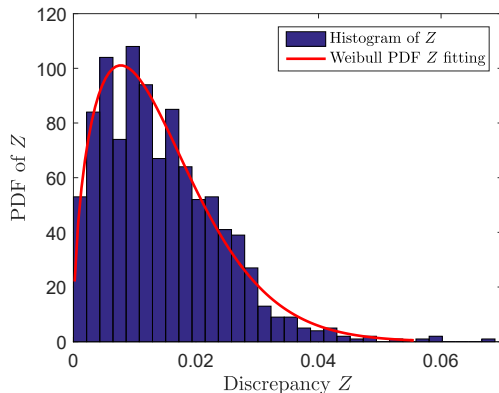


Fig. 4. The expected return $\mathbb{E}[G_{t+1}|\mathbb{F}_t]$ at $t+1$ vs. cumulative S_t with $S_t \in [0, \Theta]$ for different distributions of Z . The optimal stopping time t^* is the first time instance t when the expected return at $(t+1)$ is less than S_t .

obtained. Fig. 4 shows the expected return $\mathbb{E}[G_{t+1}|\mathbb{F}_t]$ at $t+1$ vs. S_t with $S_t \in [0, \Theta]$ for the Gamma, Weibull, Exponential and truncated Normal distributions of Z (as fitted from the real datasets in our experiments). The optimality and uniqueness of our rule is obvious where the ED immediately updates the EG with the local model when the current return at time t , $G_t = S_t$, is greater than the expected return (since from that time instance and onwards, any expected return is strictly less than any G_t , thus, we will never maximize our return). Practically, we update the model to EG when $S_t \geq S^*$, where S^* is estimated by solving (8) w.r.t. S_t ; for instance, in Fig. 4 the arrow points to the optimal discrepancy value S^* , where the ED updates EG when the current S_t exceeds this value. Note: S^* is unique, thus, the optimal stopping time t^* provided by our model is unique.

Remark 4. The PDF of Z experienced in an ED can be either fitted in a distribution function like the above-mentioned ones, that is, estimating the corresponding parameters requiring a training phase for gathering discrepancy values, or can be incrementally estimated adopting the on-line kernel Density Estimation method (Trevor et al., 2009). Based on this method, the ED incrementally updates the F_Z^t at the t th observation based on the previous F_Z^{t-1} . In both methods, the ED evaluates the criterion in (8) and optimally decides on a model update to EG.

Lemma 2. The tolerance boundary Θ , where the ED optimally delays a model update, lies in the set: $\{\Theta > 0 : 1 - \frac{\mathbb{E}[Z]}{\Theta} < F_Z(\Theta) < 1\}$. The ED decides on a delayed model update mechanism iff the expected discrepancy is strictly less than the tolerance, i.e., iff $\mathbb{E}[Z] < \Theta$. For $\Theta \leq \mathbb{E}[Z]$, the ED never delays a model update for communication efficiency. Hence, Θ is bounded in the open set $(\mathbb{E}[Z], F_Z^{-1}(1))$.

Proof. See Appendix A.5 \square

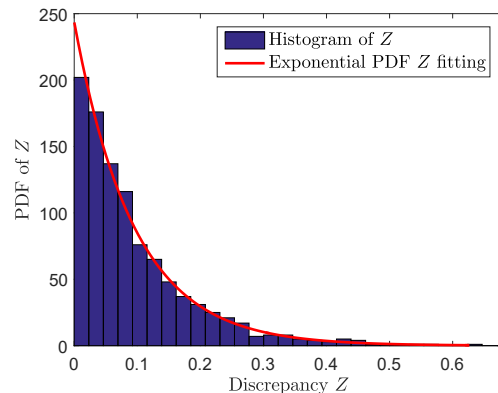


Fig. 3. (Left) Weibull distribution fitting of discrepancy Z with parameters $(\alpha_1, \beta_1) = (0.0155, 1.5147)$ for the RBF model over dataset D2; (right) Exponential distribution fitting of discrepancy Z with parameter $\mu = 0.0975$ for the LM model over dataset D2.

5.4. Adaptive decision making

The ED i has now a sequential decision making mechanism to update the EG with the model f_i based on a sequence of discrepancies since the last model update. Departing from the instantaneous decision relying only on the current discrepancy in [Harth and Anagnostopoulos \(2018\)](#), the ED i sophisticatedly postpones a model update to save communication resources and, more importantly, adapts its novelty detection mechanism in Section 4.1 based on the feedback of the discrepancy value.

The adaptive model update algorithm is provided in Algorithm 2. Firstly, the ED i receives a pair (\mathbf{x}, y) at time t . Based on the Algorithm 1, such pair is classified as novelty or familiar. In the former case, this pair is appended to the current sliding window \mathcal{N}_i (while the oldest pair is discarded) and, then, the local model f_i is either re-trained or adjusted depending on the regression algorithm. The ED i uses the locally updated f_i to instantaneously provide a prediction $\hat{y} = f_i(\mathbf{x})$, thus, obtaining the error $e_i(\mathbf{x}) = y - f_i(\mathbf{x})$. Moreover, to assess the discrepancy of the cached model f^o , the error $e_i^o(\mathbf{x}) = y - f_i^o(\mathbf{x})$ is obtained. Based on both errors, we calculate the discrepancy $z_t = |e_i - e_i^o|$ (see Lines 4–9). In instantaneous decision, we would have updated EG with the new f_i should $z_t > \theta$. However, in our mechanism, we examine whether the optimal criterion in [Theorem 4](#) is met by updating $S_t = S_{t-1} + z_t$. If the criterion in (8) holds true, the ED i updates EG with f_i and a new decision era starts off; see Lines 10–14. The decision on actually updating the cache model is delegated to the criterion in (8) to secure optimal expected return given Θ .

Algorithm 2 Model Update & Feedback Mechanism at ED i .

```

Input: input-output observed pair  $(\mathbf{x}, y)$ 
1: /*model update decision*/
2: get pair  $(\mathbf{x}, y)$  familiarity from Algorithm 1
3: if  $(\mathbf{x}, y)$  is novel (not familiar) then
4:   append  $(\mathbf{x}, y)$  in window  $\mathcal{N}_i$ 
5:   adapt/re-train model  $f_i$ 
6:   model prediction error:  $e_i(\mathbf{x}) = y - f_i(\mathbf{x})$ 
7:   (local) cached model prediction error:  $e_i^o(\mathbf{x}) = y - f_i^o(\mathbf{x})$ 
8:   error discrepancy  $z_t = |e_i(\mathbf{x}) - e_i^o(\mathbf{x})|$ 
9:   update cumulative sum:  $S_t = S_{t-1} + z_t$ 
10:  if  $S_t$  satisfies criterion in Theorem 4 then
11:    update EG with the new model  $f_i$  and  $C_i$ 
12:    the EG updates the cached model  $f_i^o \leftarrow f_i$ 
13:     $S_t \leftarrow 0$  /*start-off new decision era.*/
14:  end if
15: /*feedback*/
16: if  $z_t \leq \theta$ 
17:   penalty: relaxing  $\rho_0 = (1 + \frac{\theta - z_t}{\theta})\rho_0$  in Algorithm 1
18: else
19:   reward: shrinking  $\rho_0 = \frac{z_t - \theta}{z_t}\rho_0$  in Algorithm 1
20: end if
21: end if

```

Upon reception of a pair (\mathbf{x}, y) , the ED i is about to adjust Algorithm 1 based on the z_t value (see Lines 15–20). We exploit the occurrence of the event $\{z_t \leq \theta\}$ and the fact that the pair (\mathbf{x}, y) was classified as novelty to provide feedback. If Algorithm 1 classifies (\mathbf{x}, y) as novelty and then, after updating the model f_i (re-training or adaptation) we experience the event $\{z_t \leq \theta\}$, then we impose a penalty (negative feedback) since there was no reason to update the model and to come up with an up-to-date model with the same prediction behavior as the cached model f^o . That is, we could have saved computational resources of not proceeding with model adaptation/re-training, since regarding the prediction error, the model performance is the same as before the model adaptation. Such feedback is reflected by adjusting the error threshold ρ_0 in Algorithm 1, i.e., increasing ρ_0 by a factor $\frac{\theta - z_t}{\theta} \in (0, 1]$ to

proceed with more accurate classification results, thus, saving computational resources. On the other hand, i.e., when $\{z_t > \theta\}$ and the pair is classified as novelty, then we reward the model adaptation/re-training since we obtain expected discrepancy greater than θ between the updated f_i and cached f_i^o . In this case, we shrink ρ_0 by $\frac{z_t - \theta}{z_t} \in (0, 1)$, $z_t > \theta > 0$, to enforce model adaptation/re-training in future pairs. [Fig. 5](#) (left) shows the process at ED i including the familiarity/novelty inference and model update of Task Flow 1.

Remark 5. The advantage of Algorithm 2 is twofold. Firstly, it optimizes the decision making of when to update the local model based on [Theorem 4](#) conditioned to the non-familiarity of an input-output pair. Secondly, which is the most important functionality, it adopts a reward-penalty mechanism to adjust the error tolerance threshold ρ_0 based on the discrepancy threshold θ . Algorithm 2 provides feedback to Algorithm 1 in order to improve its novelty detection capability, which plays significant role in future local model adaptations and computational resource usage. The Algorithm 2 directly implements a closed-loop feedback controller that controls the decisions on the local model adaptation in light of saving computational resources. It achieves to increase the certainty of the Algorithm 1 on when to adapt/re-train the local model based on the feedback from the time-optimized model adaptation mechanism.

6. Expected communication

We estimate the expected communication of the instantaneous model update ([Harth and Anagnostopoulos, 2018](#)) and of delivering partially updated sufficient statistics.

6.1. Instantaneous model update communication

In the instantaneous model update ([Harth and Anagnostopoulos, 2018](#)), the ED decides on a model update or not at the t -th pair $(\mathbf{x}, y)_t$, $t = 1, \dots, T$ within $T \in \mathbb{T}$ observations. At time t , the ED only after classifying that pair as novel and re-training/adapting its local model f , accordingly, updates the EG (for updating the cached f^o with the up-to-date f) with probability $P\{Z > \theta\}$, $Z = |e(\mathbf{x}) - e^o(\mathbf{x})|$. The expected communication $\mathbb{E}[M]$ of the number of messages M sent from ED to EG within T observations is then $\mathbb{E}[M] = P\{Z > \theta\} \cdot T$. This expectation depends on the discrepancy threshold θ ; high θ is linked to low communication since ED tolerates the difference of the prediction performance between the current and cached models, however, at the expense of prediction accuracy. Low θ results to less tolerance thus high communication. Notably, if θ equals to the median m_Z of discrepancy Z , $P\{Z > \theta\} = 1/2$ and thus $\mathbb{E}[M] = T/2$. We control the expected communication by setting $\theta = \gamma m_Z$ with $\gamma > 0$.

Proposition 1. Given $\gamma > 0$ of the median of discrepancy Z between local f and cached f^o models, the expected communication of the instantaneous model update between ED and EG is $\mathbb{E}[M] = T(1 - \text{erf}(\gamma \cdot \text{erf}^{-1}(0.5)))$, where $\text{erf}(z) = 2\pi^{-1/2} \int_0^z e^{-t^2} dt$ is the error function.

Proof. See [Appendix A.6](#) □

[Fig. 5](#) (right) shows the actual and predicted $\mathbb{E}[M]$ between ED and EG as a percentage of messages w.r.t. baseline solution, where all data are transferred from ED to EG; the expected communication is accurately predicted for $\gamma \leq 2$.

6.2. Sufficient statistics update communication

The sufficient statistics C_i are conditionally adapted upon a new pair (\mathbf{x}, y) , while converge as proved in [Theorem 2](#). The adaptation, which is fundamental for convergence, results to incremental changes of the closest prototypes to pairs, thus, these changes have to be reflected to EG for model selection, as will be shown in Section 7. Let $\Delta(\mathbf{w}_\ell, u_\ell) = (\lambda\alpha\|\mathbf{x} - \mathbf{w}_\ell\|, (1 - \lambda)\text{asgn}(e - u_\ell))$ be the change vector

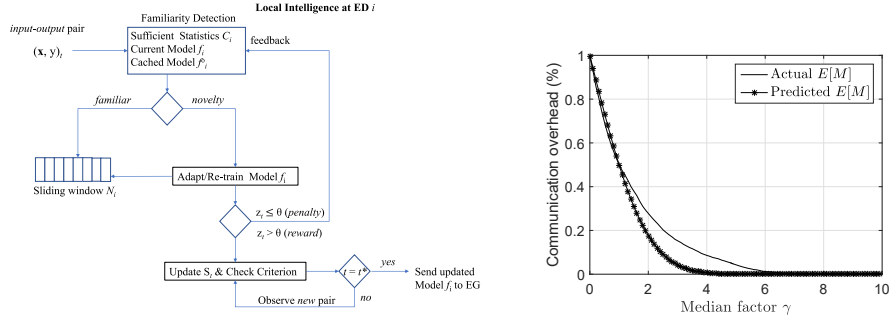


Fig. 5. (Left) The Task Flow 1 process in ED i ; (right) theoretical and experimental expected communication prediction between ED and EG vs γ in instantaneous model update.

in \mathbb{R}^{d+1} after the reception of pair (\mathbf{x}, \mathbf{y}) based on Theorem 1. After convergence, the expectation of the vector change is $\mathbb{E}[\Delta(\mathbf{w}_\ell, \mathbf{u}_\ell)] = (\mathbf{0}, 0)$; however, until convergence, the ED should regularly update EG for incremental changes in C_i . Such updates are sent from ED to EG during the update decision or can be sent interdependently, should the changes are not significant enough to be considered for update. Proposition 2 reports on the upper bound of the changes in the sufficient statistics that determines the frequency that ED updates EG considering only partially incremental updates on C_i .

Proposition 2. The expected magnitude of changes in C_i is bounded by:

$$\mathbb{E}[\|\Delta \mathbf{w}_\ell, \mathbf{u}_\ell\|] \leq 2\alpha(1 + d^{-1} + 2^{-(d+1)}) \cdot \max(\lambda \rho_I, (1 - \lambda) \rho_O).$$

Proof. See Appendix A.7 \square

The ED i determines a frequency $\propto 2\alpha(1 + d^{-1} + 2^{-(d+1)})\max(\lambda \rho_I, (1 - \lambda) \rho_O)$ for updating C_i to EG referring only to modified input/error prototypes, provided that they have not changed since the previous update.

7. Cached models selection at the edge gateway

Up to this point, we have elaborated on Task Flow 1, where ED i generates the sufficient statistics C_i to optimally update the cached model at the EG. In Task Flow 2, C_i statistics are received by EG as a guiding light to select the most appropriate diverse models per query. Our desideratum is that inferential analytics must be achieved in real-time with low communication overhead and be highly accurate. Communication overhead refers to delivery of C_i and f_i from all ED i to EG and high accuracy refers to low error for random queries.

The EG caches all models $\mathcal{F} = \{f_1^o, \dots, f_n^o\}$ received from each ED i . Based on Algorithm 2, each ED i autonomously decides when to update the EG with f_i independently of the other EDs. Partial updates of statistics C_i are also sent to EG to significantly drive model selection, as discussed in Section 6.2; EDs deliver only *knowledge* (models and statistics) to EG and *not* actual data.

Assume that analysts/applications issue a query stream $\{\mathbf{q} \in \mathbb{R}^d\}$ to Cloud, which is directed to EG; see Fig. 1. The EG should return accurate prediction $\hat{\mathbf{y}}$ and/or the relevant local models around the input space defined by query point \mathbf{q} . And, these outcomes should be highly accurate and delivered in real-time without any further communication with the EDs. Hence, given a query \mathbf{q} , the challenge for EG is to (i) efficiently select the *most appropriate* subset of models $\mathcal{F}' \subseteq \mathcal{F}$ providing an ensemble prediction (Zhou, 2012) $\hat{\mathbf{y}}$, whose prediction error is as close to the global f_G as possible and (ii) deliver the most representative models in \mathcal{F}' that better explain the input-output dependency.

7.1. On computing appropriate models subset

We show (i) that computing the best subset \mathcal{F}' of models per query \mathbf{q} is computationally hard, and (ii) that as exemplified, it is highly ben-

eficial to engage only a good models subset per query. The above showcases thus the traits and benefits of our approach on models selection at the EG. EG can, trivially, engage all cached models. Each cached model f_i^o produces an estimate $\hat{\mathbf{y}}_i = f_i^o(\mathbf{q})$ and then it takes their average $\hat{\mathbf{y}} = \frac{1}{n} \sum_{i=1}^n \hat{\mathbf{y}}_i$. Let us denote such method as the Simple Model Aggregation (SMA) as e.g., adopted in Konečný et al. (2016), so to differentiate it from EG's sophisticated methods. SMA implies that all models are equal candidates and available for providing an estimate. It would have been preferable if the EG could engage a subset $\mathcal{F}' \subset \mathcal{F}$ whose average estimate $\hat{\mathbf{y}}' = \frac{1}{|\mathcal{F}'|} \sum_{f_i^o \in \mathcal{F}'} \hat{\mathbf{y}}_i$ would be equal to $\hat{\mathbf{y}}$, or more interestingly, if the EG could engage the *minimum* subset of models whose average estimate is as close to actual \mathbf{y} than to average $\hat{\mathbf{y}}$ for each query \mathbf{q} .

Determining the minimum models subset whose aggregate estimate is close to $\hat{\mathbf{y}}$ calls to mind the Subset Sum Problem (SSP) (Przydatek, 2002): Consider a pair (\mathcal{F}, s) , where \mathcal{F} is a set of $n > 0$ positive integers and s is a positive integer. SSP asks for a subset of \mathcal{F} whose sum is closest to, but not greater than, s . SSP is NP-hard (Garey and Johnson, 1990). Consider now the following problem, referred to as Minimum Subset Average Problem (MSAP).

Problem 5. (MSAP) Given (\mathcal{F}, s) , find the minimum subset \mathcal{F}' with average s' subject to $\lfloor s' \rfloor = s$ or $\lceil s' \rceil = s$.

Theorem 5. MSAP is NP-hard.

Proof. See Appendix A.8 \square

Now, based on Theorem 5, we obtain that:

Corollary 1. Given a query \mathbf{q} , the problem of finding the minimum subset of cached models $\mathcal{F}' \subset \mathcal{F}$ in the EG, whose average estimate $\hat{\mathbf{y}}'$ gives the same error as $\hat{\mathbf{y}}$ w.r.t. the actual \mathbf{y} is NP-hard.

Proof. See Appendix A.9 \square

SSP and MSAP are NP-hard, however, one is often satisfied with an approximate, sub-optimal solution, i.e., in polynomial time; see (Przydatek, 2002) for SSP. Nevertheless, even if EG were able to use such heuristic to find the minimum set \mathcal{F}' for given query (let n be small) then this would still not be preferable given our goals. That is because, in order to obtain \mathcal{F}' for a given query, EG would *firstly* have to engage all cached models and consequently, based on their estimates, produce \mathcal{F}' . EG has to *predict* the most appropriate \mathcal{F}' , which gives the same or, hopefully, smaller prediction error than that of \mathcal{F} . This prediction can be interpreted as follows: the cached model $f_i^o \in \mathcal{F}$ might consider query center \mathbf{x} (of query \mathbf{q}) as an input observation which is deemed *unlikely* w.r.t. dataset \mathcal{X}_i of the ED i . Based on the fact that a prediction using the cached model f_i^o highly depends on the dataset \mathcal{X}_i , f_i^o will probably provide a bad estimate for \mathbf{q} w.r.t. error $e_i^o(\mathbf{x}) = \mathbf{y} - f_i^o(\mathbf{x})$. Were the EG capable of predicting the *unsuitability* of f_i^o providing a good estimate *before* engaging f_i^o then the EG could have excluded f_i^o from \mathcal{F}' .

The task of predicting \mathcal{F}' per query involves the following issues: (a) the probability (density) distribution of the queries is evidently unknown since analysts/applications randomly issue queries whose patterns are not trivially easy to be revealed and/or provided then to the EG; (b) it is not feasible to identify the distribution that generates query \mathbf{q} , since we have only one sample from this at a time; (c) it is not suitable to assume that query \mathbf{q} is produced by a certain distribution at time t , which remains also the same for subsequent queries $\mathbf{q}_\tau, \tau > t$. This is getting more difficult when dealing with non-stationary distributions of query patterns, which is not a rare situation (Anagnostopoulos and Triantafyllou, 2017a). The only available knowledge we can exploit for predicting an appropriate models subset per query is the set of the sufficient statistics $\{C_i\}_{i=1}^n$ delivered (and updated) from the EDs to their EG. Based on such knowledge, we propose computationally efficient and accurate model selection algorithms for edge-centric inferential analytics.

7.2. Cached model selection algorithms

We introduce model selection methodologies exploiting knowledge coming from EDs. The ensemble prediction \hat{y} is the weighted sum of $\hat{y}_i = f_i^o(\mathbf{q})$:

$$\hat{y} = \sum_{i=1}^n f_i^o(\mathbf{q}) \beta_i(\mathbf{q}). \quad (9)$$

The weight $\beta_i(\mathbf{q})$ in (9) is a function of the current query \mathbf{q} that interprets the importance of the performance of local model f_i in the local familiar input subspace around query \mathbf{q} derived by the sufficient statistics C_i . The $\beta_i(\mathbf{q})$ value drives the definition of $\mathcal{F}' \subseteq \mathcal{F}$ where EG engages *only* the models in \mathcal{F}' for this specific query. The baseline solution is the SMA, which does not exploit the statistics C_i in the ensemble outcome, i.e., the EG simply aggregates the individual predictions $\hat{y}_i = f_i^o(\mathbf{q})$ for deriving the final one thus setting $\beta_i(\mathbf{q}) = 1/n$: $\hat{y} = f_{\text{AVG}}(\mathbf{q}) = \frac{1}{n} \sum_{i=1}^n \hat{y}_i$. In SMA the EG is *only* updated independently by an ED i with f_i , while no reception of C_i is required by any ED. The ensemble subset $\mathcal{F}' \equiv \mathcal{F}$, i.e., no model selectivity, where prediction accuracy is not favored compared to global f_G ; see evaluation Section 8. We now propose the following model selection methodologies departing from SMA.

7.2.1. Input-space aware top- \mathcal{K} model (IAM)

We first present the top-1 (best) model selection scheme ($\mathcal{K} = 1$). The EG selects only one (best) model $f^* \in \mathcal{F}$ to engage analytics tasks, i.e., $\mathcal{F}' = \{f^*\}$ given query \mathbf{q} . The model selection is achieved by using the input space prototypes $\{\mathbf{w}_{i,k}\}$ of the sufficient statistics C_i received at EG. It is worth mentioning here that the input prototypes $\mathbf{w}_{i,k}, \forall i$ are *dragged* to the subspaces of the input-error space to reflect the prediction performance of the considered model f_i (please, recall Remark 3). In this rationale, the IAM selects the model f^* whose the ℓ -th input prototype \mathbf{w}_{ℓ}^* is the closest to query \mathbf{q} compared to *all* input prototypes in $\mathcal{W} = \{\{\mathbf{w}_{1,k}\}_{k=1}^{k_1} \cup \dots \cup \{\mathbf{w}_{n,k}\}_{k=1}^{k_n}\}$ from *all* n models: $\mathbf{w}_{\ell}^* = \arg \min_{\mathbf{w} \in \mathcal{W}} \|\mathbf{q} - \mathbf{w}\|$. The EG selects f^* whose input subspace (represented by \mathbf{w}_{ℓ}^*) is the most familiar (closest) with query point \mathbf{q} , thus, the associated predictive model f^* can provide the best prediction. Without having obtained all input prototypes \mathcal{W}_i from each f_i , the EG could not discriminate which model's input subspace is the most familiar with the given query point. The weight function in IAM indicates the closest distance of \mathbf{q} to \mathbf{w}_{ℓ}^* :

$$\beta_i(\mathbf{q}) = \begin{cases} 1 & \text{if } \exists \mathbf{w}_{i,k} \in \mathcal{W}_i : \mathbf{w}_{i,k} = \mathbf{w}_{\ell}^* \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

The EG engages only the f^* associated with the closest prototype for prediction, i.e., $\hat{y} = f^*(\mathbf{q})$. For $\mathcal{K} > 1$, the EG ranks all prototypes $\mathbf{w} \in \mathcal{W}$ w.r.t. their distance from query \mathbf{q} and selects those models

$f_1^*, \dots, f_{\mathcal{K}}^* \in \mathcal{F}' \subset \mathcal{F}$ whose closest input prototypes are ranked in the top- \mathcal{K} closest distances. The ensemble prediction is then:

$$\hat{y} = \sum_{i=1}^{\mathcal{K}} f_i^*(\mathbf{q}) \beta_i^*(\mathbf{q}), \quad (11)$$

where $\beta_i^*(\mathbf{q})$ is normalized to $[0,1]$ w.r.t. the top- \mathcal{K} inverse distances:

$$\beta_i^*(\mathbf{q}) = \frac{e^{-\|\mathbf{q} - \mathbf{w}_{i,\ell}^*\|^2}}{\sum_{l=1}^{\mathcal{K}} e^{-\|\mathbf{q} - \mathbf{w}_{l,\ell}^*\|^2}}. \quad (12)$$

The influence of the distance $\|\mathbf{q} - \mathbf{w}\|$, i.e., the closer to \mathbf{q} the higher the weight importance, is achieved by the exponential inverse squared distance weighting $e^{-\|\mathbf{q} - \mathbf{w}\|^2}$ (Lukaszyk, 2004). Note, for $\mathcal{K} = n$, we obtain the Weighted SMA, where the normalized weights reflect the distance of query \mathbf{q} to the *local* closest prototypes from each model.

7.2.2. Error-aware top- \mathcal{K} model (EAM)

The EAM model combines \mathcal{K} cached models from \mathcal{F} under the double-exponential error weighting function $g(u) = 0.5e^{-|u|}, u \in \mathbb{R}$. The information of local errors per model is provided by the corresponding input-error associations and specifically from the error representatives. A model f_i^o is appropriately weighted according to the assessment of predictions in terms of average absolute prediction error $u_{i,k}$ around the input sub-space $\mathbf{w}_{i,k}$ such that $\mathbf{w}_{i,k} = \arg \min_{l \in [K_i]} \|\mathbf{q} - \mathbf{w}_{i,l}\|$. Specifically, given a query \mathbf{q} we derive the closest input prototype $\mathbf{w}_{i,k}$ and its associated error prototype $u_{i,k}$. Then, by adopting the double-exponential density $g(u)$ over the average absolute prediction errors represented by the prototypes $u_{i,k}$, we obtain:

$$\beta_i(\mathbf{q}) = \frac{g(u_{i,k})}{\sum_{l=1}^n g(u_{l,k})} : \mathbf{w}_{i,k} = \arg \min_{m \in [K_i]} \|\mathbf{q} - \mathbf{w}_{i,m}\|. \quad (13)$$

The prediction outcome is achieved by selecting $\mathcal{K} \geq 1$ models from \mathcal{F} with the top- \mathcal{K} weights $\beta_i(\mathbf{q})$ of the \mathcal{K} models, i.e., $\hat{y} = \sum_{i=1}^{\mathcal{K}} f_i(\mathbf{q}) \beta_i(\mathbf{q})$; $\beta_i(\mathbf{q})$ in (13).

7.2.3. Input & error-space aware top- \mathcal{K} model (IEAM)

The EG exploits all the knowledge from $C_i, \forall i$ combining the familiarity of the input subspace of a model w.r.t. query \mathbf{q} through the closest input prototype $\mathbf{w}_{i,\ell}$ and the associated performance reflected by the error prototype $u_{i,\ell}$. IEAM selects the best or the top- \mathcal{K} best models from \mathcal{F} , which are not only familiar w.r.t. the queried input but *also* effective for providing accurate predictions based on their local prediction performance over the familiar subspace represented by the closest input prototypes to the query point. The combination of the two directions, input space familiarity and associated prediction performance, renders the EG to proceed with a more sophisticated model selection. The weight $\beta_i(\mathbf{q})$ represents a *degree of model closeness* to an issued query taking into consideration the (inverse) closest input distance $\mathbf{w}_{i,\ell} \in \mathcal{W}_i$ and the associated median of the absolute prediction error $u_{i,\ell}$ around this subspace. Specifically, $\beta_i(\mathbf{q})$ interprets the relative closeness of model f_i to query \mathbf{q} :

$$\beta_i(\mathbf{q}) = \frac{e^{-\|\mathbf{q} - \mathbf{w}_{i,\ell}\|^2} (1 - \bar{u}_{i,\ell})}{\sum_{l=1}^{\mathcal{K}} e^{-\|\mathbf{q} - \mathbf{w}_{l,\ell}\|^2} (1 - \bar{u}_{l,\ell})}, \quad (14)$$

where $\bar{u}_{i,k} = \frac{u_{i,k}}{\sum_{u \in \mathcal{U}} u}$ is the normalized median of the prediction error of model f_i over the k -th input/error subspace among all error medians $\mathcal{U} = \{\{u_{1,k}\}_{k=1}^{k_1} \cup \dots \cup \{u_{n,k}\}_{k=1}^{k_n}\}$ from *all* n models. The prediction outcome is achieved by selecting $\mathcal{K} \geq 1$ models from \mathcal{F} with the top- \mathcal{K} high degrees of closeness of the \mathcal{K} models ranked by $\beta_i(\mathbf{q})$, i.e., $\hat{y} = \sum_{i=1}^{\mathcal{K}} f_i(\mathbf{q}) \beta_i(\mathbf{q})$; $\beta_i(\mathbf{q})$ in (14).

7.3. Computational & space complexity at the edge

We provide the inherent computational complexity of our methodology including EDs and EGs trading off communication efficiency and accuracy of analytics. The computational complexity of the Algorithm 1 in Task Flow 1 is as follows. Given (\mathbf{x}, \mathbf{y}) the ED i adopts a d -dim. tree structure over the K_i prototypes to classify the pair as familiar in $O(d \log K_i)$ time. The decision on a model adaptation due to a novelty classification with probability $\xi_i < 1$ is then $O(d)$ (see Appendix B). In the context of model adaptation, the complexity is now based on the underlying model algorithm. Should the algorithm be incremental, the complexity for adaptation is upper bounded by $O(dN)$, given a sliding window of N d -dim. vectors. However, there are incrementally updated algorithms whose complexity is significantly reduced and does not directly depend on the window size N . The selection of incrementally updated models in an ED is evidently guided by the computational capabilities and resource availability of the ED. For instance, in a resource constrained environment, EDs normally adopt regression/classification models with complexity $O(d)$ in an on-line adaptation mode, like online passive-aggressive algorithms or stochastic-gradient decent RLS (Crammer et al., 2006) (see Appendix B for model update adaptation complexity). Algorithm 2 incorporates the complexity of the model update criterion in Theorem 4, which is $O(1)$ adopting incremental KDE with space complexity $O(dN)$ given the sliding window of N d -dim. vectors. The feedback mechanism computes in $O(1)$ the prediction error, since the input-output pair is observed in ED i . Hence, in total, the model update and adaptation overhead in ED i to compute and decide is $O(d \log K_i) + \xi_i O(d + 1)$ time, which is a significantly light process given the current advances on ED technology. The expected communication overhead for the model update and knowledge transfer to the EG is analyzed in Proposition 1. The space complexity of such knowledge transfer of ED i incorporates the parameters of the adapted f_i model and potential changes/deltas of input-error prototypes $\{\Delta \mathbf{w}_k, \Delta u_k\} \in C_i$. The space complexity is $O(K_i(d + 1))$ given that all prototypes have been changed.

In Task Flow 2, given a query \mathbf{q} , the EG performs one nearest neighbor (1NN) search over the input prototypes in all statistics $\{C_i\}_{i=1}^n$ to find the closest one used for all the proposed model selection schemes. By adopting a d -dim. tree over the $n' = n \times \sum_{i=1}^n K_i$ prototypes, the time complexity per query is $O(d \log(n'))$ to provide inferential analytics with $O(dn')$ space complexity.

7.4. Discussion on data & predictive task offloading at the edge

An essential mechanism in EC is the computation task and data offloading as a process of delegating computation tasks and/or data to an EC server or Cloud (Alghamdi et al., 2019a). As emerging predictive applications require intensive computation processes, task/data offloading is a promising mechanism that potentially overcomes limitations of EDs. There are three fundamental offloading decisions: local task execution/data processing; full task/data offloading; and partial offloading (Alghamdi et al., 2019b), where in the latter part of the computation is processed locally while the rest is offloaded to EC server or Cloud. EDs and EGs in our context have tasks, e.g., model training; inference; adaptation; updates, and data to be potentially offloaded to guarantee quality of service. Therefore, EDs can autonomously decide on suitable decisions based on current context, e.g., computational resources and load, to secure resilience in delivering predictive tasks (Alghamdi et al., 2019c). In our scheme, the model update mechanism is timely-optimized to offload meta-data, i.e., model parameters and statistical prototypes, from EDs to EGs to achieve knowledge aggregation. Our future agenda includes sophisticated task/data/knowledge offloading mechanisms balancing the expected workload among EDs and EGs w.r.t. load and computational capacity of EDs in resource-constrained environments. Notably, our evaluation section offers comprehensive

evaluation results over unmanned vehicles, which is undoubtedly considered as a resource constrained environment.

8. Experimental evaluation

8.1. Datasets, predictive models & parameters

We experiment with real multivariate contextual datasets from EDs including stationary sensors and unmanned vehicles. For each dataset, we define scenarios corresponding to regression and time series forecasting adopting multivariate linear regression (LM), autoregressive (AR) recursive least-squares embedding (AR-RLS), radial basis function network regression (RBF), and nonlinear AR-RBF, which are widely used for analytics (Babcock et al., 2002; Anagnostopoulos and Triantafillou, 2017a, 2017b; Tatbul et al., 2003). The reader could skip this sub-section, should they be familiar with such predictive models.

Regression Models: In LM, an ED i learns the model $y = f_i(\mathbf{x}) = \mathbf{b}_i^T \mathbf{x}$ with parameter $\mathbf{b}_i \in \mathbb{R}^d$. The RBF model has an input and a hidden layer with a non-linear RBF function h and a linear output layer. The ED i learns the RBF model: $y = f_i(\mathbf{x}) = \sum_{m=1}^{M_i} a_{im} h(\|\mathbf{x} - \boldsymbol{\mu}_{im}\|)$, with parameters $B_i = \{a_{im}, \boldsymbol{\mu}_{im}\}_{m=1}^{M_i}$. M_i is the number of neurons in the hidden layer, $\boldsymbol{\mu}_{im}$ is the centroid vector for neuron m , and a_{im} is the weight of neuron m in the linear output. The $h(\cdot)$ depends on the input distance from a centroid and is radially symmetric about the centroid commonly used to be Gaussian, i.e., $h(\|\mathbf{x} - \boldsymbol{\mu}_{im}\|) = \exp(-0.5\|\mathbf{x} - \boldsymbol{\mu}_{im}\|^2)$. The B_i are sequentially estimated optimizing the fit between f_i and the pairs (\mathbf{x}, \mathbf{y}) . Appendix B provides the incremental rules for the models' parameters.

Time-series Forecasting Models: For the time series forecasting, we adopt the embedding mechanism (Cerqueira et al., 2017) to transform the time series and then apply AR-RLS and AR-RBF. Consider the time series of values u_1, u_2, \dots, u_t at regular time intervals. The time series is reconstructed into a higher dimensional space with embedding dimension d by generating a matrix of N embedding vectors in d dimensions: $\mathbf{A}_{N,d} = [\mathbf{u}_1^T; \dots; \mathbf{u}_{N+1}^T]$, each row corresponding to an embedding vector $\mathbf{u}_t = [u_t, u_{t-1}, \dots, u_{t-d+1}]^T \forall t \in \mathbb{T}$. A slides with the sliding window \mathcal{N}_i of size N of the ED i . Based on this transformation, there are no long term time dependencies in the series, thus, the embedding vectors are deemed as uncorrelated (Takens, 1981) and allows the use of any regression technique in the literature; in our experiments is AR-RLS and AR-RBF and obtain the pair $(\mathbf{x}, \mathbf{y})_t$ from embedding vector \mathbf{u}_t with $y_t = u_t$ and $\mathbf{x}_t = [u_{t-1}, u_{t-2}, \dots, u_{t-d+1}]^T \in \mathbb{R}^{d-1}$. The forecasting task is to predict $y_{t+p} = u_{t+p}$ at time $t + p$, with lag $p > 1$, given the $d - 1$ recent values $\mathbf{x}_t = [u_{t-1}, \dots, u_{t-d+1}]$ and N recent embedding vectors $\mathbf{u} \in \mathbf{A}$ matrix. In AR-RLS, we obtain: $y_t = u_t = \sum_{j=1}^{d-1} b_j u_{t-j}$, where AR coefficients $\mathbf{b} = [b_1, \dots, b_d]^T \in \mathbb{R}^d$ are incrementally updated based on RLS after receiving a new pair $(\mathbf{x}, \mathbf{y})_t = \mathbf{u}_t = (u_t, u_{t-1}, \dots, u_{t-d+1})$. In AR-RBF, we obtain: $y_t = u_t = \sum_{m=1}^{M_i} a_{im} h(\|\mathbf{x}_t - \boldsymbol{\mu}_{im}\|)$. B reports on the incremental parameters estimation for all models.

Intel Lab Dataset (D1): D1 dataset⁵ assigns two EGs with $n = 25$ EDs each. Each ED captures 3-dim. ($d = 3$) vectors of temperature, humidity and light (2.3 million values for each in 36 days) every 31s. We experiment with the LM and RBF built and maintained by the EDs. Each ED i learns a LM $y = f_i(\mathbf{x})$ with input $\mathbf{x} = [x_1, x_2]^T$, x_1 is temperature, x_2 is humidity and output y is light and parameter $\mathbf{b}_i \in \mathbb{R}^3$ (including the offset coefficient) and a RBF with inputs and output as described for the LM and parameters $B_i = \{a_{im}, \boldsymbol{\mu}_{im}\}_{m=1}^M$; $M = 11$ is experimentally determined. The f_i 's are adjusted over sliding window of $N = 120$ vectors (1 h history) and incrementally updated (Appendix B).

Gas Sensor Array Drift Dataset (D2): D2 dataset⁶ consists of measurements collected by an array of $n = 16$ EDs/chemical sensors resulting in 7500 $d = 6$ dim. vectors per ED referring to six gases at various

⁵ <http://db.csail.mit.edu/labdata/labdata.html>.

⁶ <https://archive.ics.uci.edu/ml/datasets/gas+sensor+array+drift+dataset>.

levels of concentrations. For regression, we predict the gas concentration Amonia (output y) based on five gas attributes (input x) per ED. The EDs learn 6-dim. LM and RBF models estimating the $b_i \in \mathbb{R}^6$ and B_i parameters, respectively. In RBF, the number of neurons $M = 7$ is empirically obtained from analyzing the data. The f_i 's are incrementally adjusted over sliding window of $N = 50$ vectors (Appendix B). For time series forecast, we predict the future value of each gas attribute at $t + 1$ given a recent history of values using AR-RLS and AR-RBF with $d = 5$ dim. embedding vectors (the embedding $d = 5$ is estimated by AIC; Appendix B). The EDs learn for each gas $j \in \{1, \dots, 6\}$, a $d = 5$ -dim. AR-RLS and AR-RBF with $b_{ij} \in \mathbb{R}^5$ and B_{ij} ($M_{ij} = 4$), respectively, for forecasting the j th gas. The f_i 's are incrementally adjusted over $N = 22$ embedding vectors.

Unmanned Surface Vehicles Dataset (D3): D3 dataset⁷ comprise mobile sensor readings from a swarm of $n = 4$ Unmanned Surface Vehicles (USVs; three PlaDyPos⁸ USVs and one MST⁹ USV) from the GNFUV project¹⁰. The swarm moves w.r.t. GPS pre-defined trajectories and the USVs are floating over the sea surface in a coastal area of Athens (GR) sensing ($d = 2$) humidity and temperature of the sea surface. Each USV represents a ED capturing 1672 2-dim. vectors every 10 s and is equipped with a Raspberry Pi for locally computing predictive models and obtaining wireless access to a EG; a testbed framework provided by the RAWFIE project.¹¹ For regression, we monitor the temperature-humidity correlation on the sea surface captured by the swarm with output y (temperature) and input x (humidity) per ED/USV. The EDs learn LM and RBF models with $b_i \in \mathbb{R}^3$ and B_i ($M = 6$), respectively. The models are incrementally updated over window size $N = 6$ (1 min history). For time series forecast, we predict the future temperature and humidity using AR-RLS and AR-RBF with $d = 6$ dim. embedding vectors (estimated by the AIC (Burnham and Anderson, 2002)). The AR-RLS and AR-RBF parameters $b_i \in \mathbb{R}^6$ and B_i with $M_i = 3$, respectively, incrementally adjusted over window size $N = 6$.

Parameters: In Algorithm 1, learning rate $\alpha = 0.1$ (Bottou and Bousquet, 2007) and regularization factor $\lambda = 0.5$ in (3) for putting equal importance of EQE and EPE. The familiarity threshold ρ_i is normalized in the input domain $[0, 1]^d$, i.e., $\rho_i/\sqrt{d} \in (0, 1)$; a value close to 1 refers to coarse vector quantization, thus, a few prototypes K , while close to 0 refers to fine-grained quantization, thus many prototypes K . For models update, in each ED i the discrepancy threshold $\theta_i = \gamma MED_i$ with factor $\gamma \in (0, 3]$ and MED_i is the median of the error differences $|e_i(x) - e_i^o(x)|$ in Algorithm 2 to control the expected communication between ED and EG. Based on θ_i , the initial error tolerance $\rho_O = \theta_i$ with minimum $\rho_O^* = \frac{\theta_i}{20}$. The boundary $\Theta \in \{2\theta, \dots, 10\theta\}$ thus being proportional to the median of discrepancy z . Table 1 summarizes the parameters ranges/values used in our experiments.

8.2. Performance metrics

The performance metrics reflect the diverse objectives of this research. Firstly, regarding the time-optimized model update mechanism, we assess the *expected maximum return* $\mathbb{E}[G_{t^*}]$, when ED decides on model update at optimal stopping times t^* compared to Θ . A close expected return to Θ indicates that the ED intelligently decides on a model update without exceeding such boundary, thus, enjoying less communication overhead. In terms of communication overhead, we assess the *update rate* of the proposed mechanism defined as $\frac{1}{\mathbb{E}[t^*]}$ given a specific time horizon T . We desire our mechanism to decrease the

redundant model updates, however, not exceeding the tolerance boundary and not spoiling the quality of analytics at the EG. We define as *consistency* the ratio $\frac{\mathbb{E}[S_{t^*}]}{\mathbb{E}[t^*]}$ indicating the average discrepancy at the EG due to model updates. Such metric indicates that our time-optimized mechanism compared with (Harth and Anagnostopoulos, 2018) and other model update mechanisms (provided later), on average, result to the same discrepancy at the EG. Hence, the proposed mechanism does not spoil the quality of analytics at the EG in light of reducing the communication overhead. Instead, as we will show, our mechanism *knows when to update* the model at the ED enjoying the same discrepancy levels with other model update mechanisms being communication efficient. The consistency and the expected return metrics showcase the optimality of our scheme for model updates. We also define the *percentage of the expected EDs-EG communication savings* of all model selection schemes compared to the Global/baseline approach; recall that the baseline approach sends all raw data from EDs towards the EGs to construct a global model f_G . In our case, we measure the communication overhead for delivering models and sufficient statistics from the EDs to EG. Finally, concerning the *prediction accuracy* of the inferential analytics per query, we measure the Root Mean Squared Error: $RMSE = \left(\frac{1}{L} \sum_{l=1}^L (\hat{y}_l - y_l)^2\right)^{1/2}$ and the Mean Absolute Error: $MAE = \frac{1}{L} \sum_{l=1}^L |\hat{y}_l - y_l|$ over L regression queries $\{q_l\}_{l=1}^L$ issued from the applications/analysts to the EG. The queries were generated by splitting the datasets into *test* and *training* set using 10-fold-cross validation (Trevor et al., 2009).

8.3. Comparison schemes & rules

Predictive Model Update Rules: Under the philosophy of finding the best time to update the models from EDs to the EG, we compare our mechanism with two model update mechanisms: Firstly, Harth and Anagnostopoulos (2018) considers the rule: $t^* = \min\{t > 0 : z_t = |e_t(x) - e_t^o(x)| > \theta\}$, hereinafter referred to as Instantaneous Rule (INST). We also consider an update rule where ED updates EG when the average discrepancy since the last update is greater than θ : $t^* = \min\{t > 0 : \frac{1}{t} \sum_{\tau=1}^t z_\tau > \theta\}$, hereinafter referred to as the Mean Rule (MEAN). MEAN might resemble at the first sight with our Optimal Stopping Time Rule (OST). However, it does only consider constraints over the current mean discrepancy and does not optimize the expected return. MEAN is rather intuitive, nonetheless, without optimizing the distance from any tolerance threshold or delaying the model update for reducing the communication overhead.¹² **Note:** as proved in Theorem 4, the optimality of our OST is guaranteed, thus, any other model update mechanism does not maximize (6).

Schemes under Comparison: We compare our approach with schemes found in the literature. DBP (Raza et al., 2015) uses forecasting models to predict ED's data, one forecasting model per attribute independently, and compares the predicted values with the current ones. If the difference less than a tolerance then DBP remains idle; otherwise, DBP builds a new forecasting model per attribute and transmits the models and data to EG. In DBP, the model selection scheme at the EG is the SMA. HOVF (Harth and Anagnostopoulos, 2017) uses an AR model for local ED's data prediction per attribute, independently. HOVF decides whether to send only data to EG or not based on a stopping time mechanism in Shiryayev (2008). Both HOVF and DBP require EG to reconstruct the data (if they are not transmitted from the EDs to EG) in order to build the models f_i at the EG, while the model selection scheme in HOVF is the SMA. Our previous work (Harth and Anagnostopoulos, 2018) employs model update by adopting INST, model selection using

⁷ <http://archive.ics.uci.edu/ml/datasets/GNFUV+Unmanned+Surface+Vehicles+Sensor+Data>.

⁸ <http://pladyfleet.fer.hr/>.

⁹ <http://www.oceanscan-mst.com/>.

¹⁰ <https://sites.google.com/view/gnfuv/home>.

¹¹ <http://www.rawfie.eu/>.

¹² The interested reader could refer to (Shiryayev, 2008) where there the optimal stopping time for maximizing the average S_t/t (considering no constraints) is intractable, even for specific simple cases of expectations of Z .

Table 1
Experimental parameters & ranges/values.

Parameter	Notation	Value/Range
Dataset		D1, D2, D3
Predictive models	f	LM, RBF, AR-RLS, AR-RBF
Dimension	d	$\{2, \dots, 6\}$
#EDs per EG	n	$\{4, 16, 25\}$
Sliding window size	N	$\{6, 22, 50, 120\}$
# RBF neurons	M	$\{3, 4, 6, 7, 11\}$
Learning rate	α	0.1
Regularization factor	λ	0.5
Familiarity threshold	ρ_f/\sqrt{d}	$[0,1]$
Error median factor	γ	$(0,3]$
Discrepancy threshold	θ	$\gamma \cdot MED$
Error tolerance	ρ_O	$\geq \frac{\theta}{20}$
Discrepancy tolerance	Θ	$[2\theta, 11\theta]$
#Regression queries	L	3000
Penalty factor	B	Crammer et al. (2006), Anagnostopoulos and Triantafillou (2017b) and Jain and Tata (2017)

IEAM (Section 7.2) and the statistics update rule in Section 6.2, thus, obtaining the scheme: INST + IEAM. In this paper, our scheme variants are: OST + SMA, OST + IEAM, OST + IAM, and OST + EAM, i.e., ED uses OST in Theorem 4 and EG uses either SMA, EAM, IAM, or IEAM (SMA is obtained in Konečný et al. (2016)). Finally, all schemes are compared against the Global scheme, where no models are obtained, no model selection is provided and only data are transferred to EG to build the global f_G . The window size and tolerance for DBP and HOVF are the same as in OST for the sake of comparison.

8.4. Performance & comparative assessment

We assess our hypothesis on the optimality of OST compared to INST (Harth and Anagnostopoulos, 2018) and MEAN in terms of expected return $\mathbb{E}[G_{t^*}]$, expected model update rate $\frac{1}{\mathbb{E}[t^*]}$, expected communication $\mathbb{E}[M]$ and consistency $\frac{\mathbb{E}[S_{t^*}]}{\mathbb{E}[t^*]}$. Fig. 6 shows the PDF of stopping times for all update rules. OST optimally delivers model *only when* it is deemed appropriate, thus, saving communication resources achieving same or even higher analytics quality compared to MEAN and INST shown in Fig. 7 (left). INST and MEAN result in spontaneous & redun-

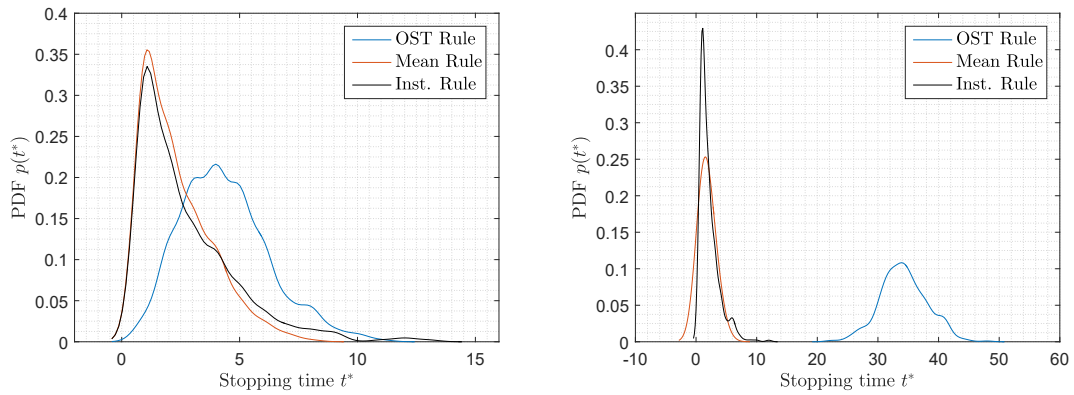


Fig. 6. PDF of stopping times t^* for OST, INST, and MEAN over D3 dataset with $\Theta = 5\theta$ using (left) AR-RLS and (right) AR-RBF.

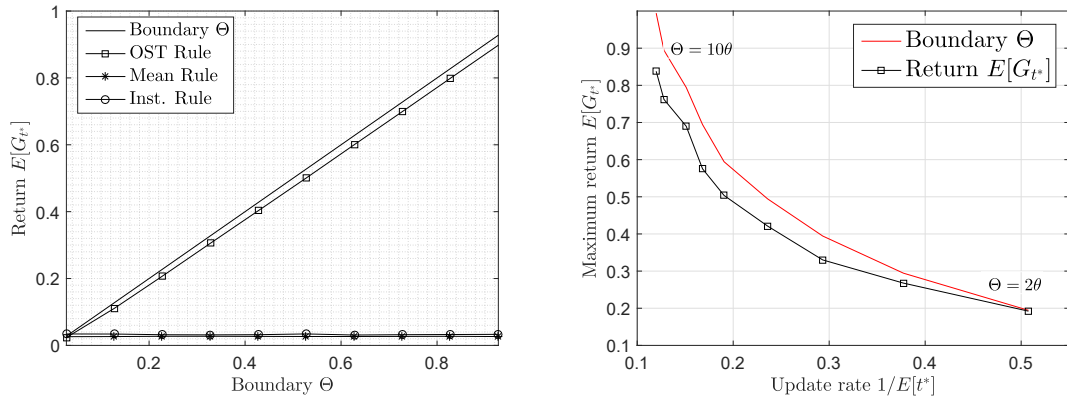


Fig. 7. Expected maximum return $\mathbb{E}[G_{t^*}]$ (left) vs. boundary Θ for OST, INST, and MEAN; (right) vs. update rate $\frac{1}{\mathbb{E}[t^*]}$ for different Θ for OST (D2 dataset, AR-RBF model).

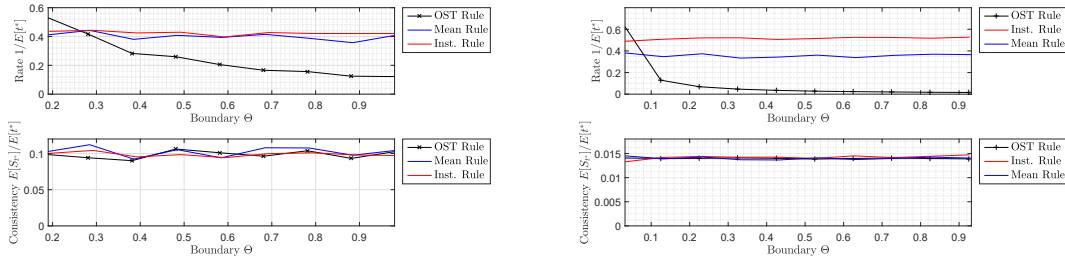


Fig. 8. (Left/right:upper) expected update rate $\frac{1}{E[r^*]}$ and (left/right:lower) consistency vs. boundary Θ for OST, MEAN and INST; D3 dataset, (left) AR-RLS and (right) AR-RBF.

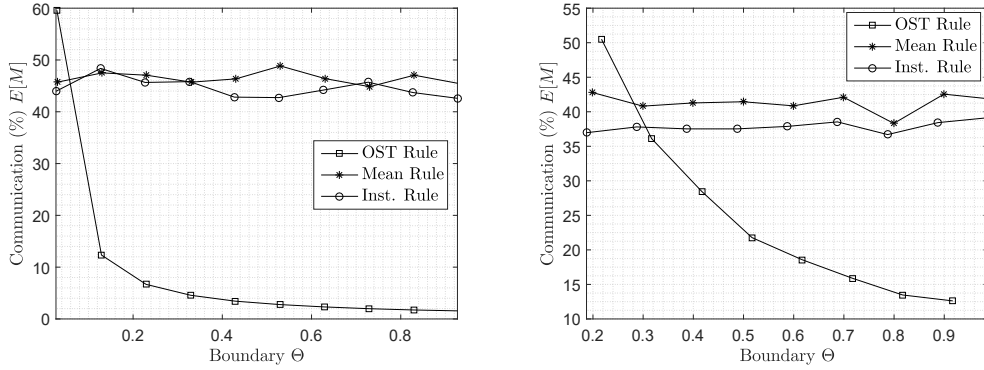


Fig. 9. Expected communication percentage (w.r.t. Global) $E[M]$ vs. boundary Θ for OST, MEAN and INST over D1 dataset using (left) LM and (right) RBF models.

dant updates especially with accurate predictive AR-RBF obtaining high distance of the expected discrepancy from Θ , while OST intelligently delays as much as possible any update being closest Θ shown in Fig. 7 (left). Similar results are obtained using D1 and D2 with all predictive models; not shown due to space limitations. Fig. 7 (right) shows the optimality of OST achieving a relatively close expected return to every Θ value trading-off analytics quality vs. update rate. Given low Θ , OST increases the update rate by maximizing the expected quality and being closest to Θ ; this is not achieved by any other rule. Fig. 8 (left/right:lower) shows the consistency of all rules indicating that the EG experiences the same discrepancy of analytics quality adopting OST, while OST being significantly more communication efficient as shown in Fig. 9. In Fig. 9 the communication percentage for OST decreases with Θ reaching less than 15% while being as consistent as the other rules and INST and MEAN achieve communication around 42%; similar results obtained for D2 & D3 for all predictive models.

We now assess and compare our scheme with the above-mentioned comparison w.r.t. communication efficiency and accuracy of inferential analytics at the edge. We assess our hypothesis where knowing the best local model f_i to involve at the EG per query q is unknown since

we cannot know if $q \in \mathcal{X}_i$. In terms of model selection, Fig. 10 (left) shows the MAE differences Δe_i of IAM, EAM, IEAM, SMA (Konečný et al., 2016) and Global compared to the *known* best f_i per ED i , i.e., the ideal case knowing which model to engage. Using IEAM and EAM, 44% of the cases obtain similar accuracy with the Global, IAM achieves same accuracy in 20% of the cases, while SMA obtains only 6% of the cases similar accuracy to Global resulting to the highest difference thus highly inappropriate for model selection. This indicates the capability of

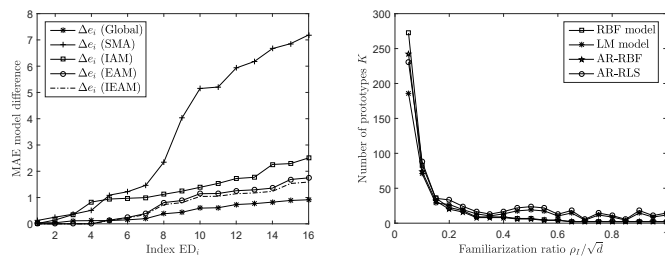


Fig. 10. (Left) MAE differences for $n = 16$ EDs w.r.t. ideal case, i.e., knowing the best model to engage (D2 dataset, LM model); (right) #prototypes K vs. ρ_l/\sqrt{d} (D3 dataset, $d = 2$ for LM and RBF and embedding $d = 6$ for AR-RLS and AR-RBF); $\Theta = 50$.

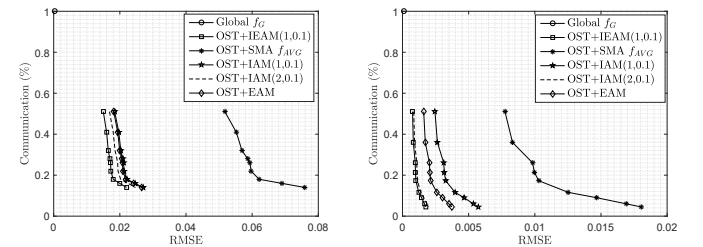


Fig. 11. Expected communication ratio (w.r.t. Global scheme) for all OST scheme variants vs. RMSE for different Θ over D2 dataset using (left) AR-RLS and (right) AR-RBF models.

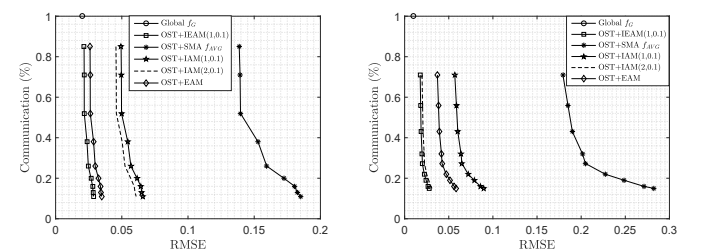


Fig. 12. Expected communication ratio (w.r.t. Global scheme) for all OST scheme variants vs. RMSE for different Θ over D3 dataset using (left) LM and (right) AR-RBF models.

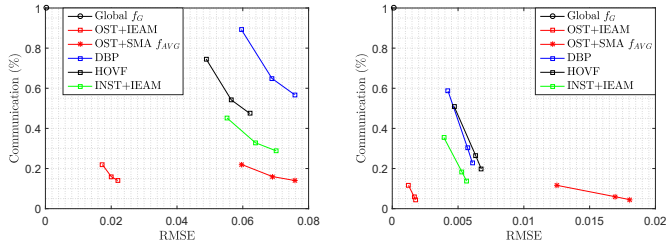


Fig. 13. Expected communication ratio for all comparison schemes vs. RMSE for $\Theta \in \{50, 70, 100\}$ over D2 dataset using (left) LM and (right) RBF models.

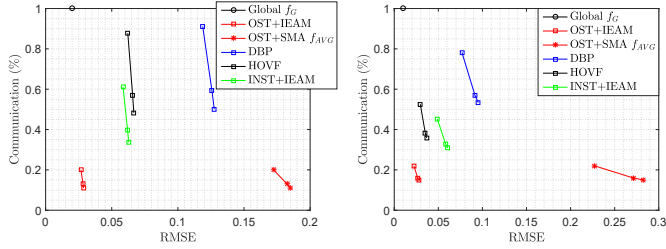


Fig. 14. Expected communication ratio for all comparison schemes vs. RMSE for $\Theta \in \{50, 70, 100\}$ over D3 dataset using (left) AR-RLS and (right) AR-RBF models.

IEAM and EAM to identify the most appropriate local models *per query* at EGs without raw data transfer thus being communication efficient and generating as accurate predictions as the Global. We obtain similar results for D1 ($n = 25$) and D3 ($n = 4$); not shown for space limitations. Fig. 10 (right) shows the average number of prototypes K per ED vs. ratio ρ_l/\sqrt{d} for all predictive models; ratio towards 1 decreases K being negative exponential indicating the minimum storage requirement on EDs retaining prototypes for achieving accurate predictions as Global without transferring data ($\rho_l/\sqrt{d} = 0.1$ obtains $K = 73$ per ED). To illustrate the *efficiency* of our scheme variants trading-off RMSE with communication, Fig. 11 and Fig. 12 show a significant 50–88% decrease in communication for OST + IEAM/IAM/EAM (top $K \in \{1, 2\}$ models and $\rho_l/\sqrt{d} = 0.1$), which achieves RMSE slightly higher than Global for all Θ starting from 2θ to 10θ (top left to right bottom). Note: the increase of RMSE and communication reduction in OST variants except OST + SMA are not highly correlated, as the error with high communication results is nearly the same error than with nearly no communication. This indicates that EG identifies the best models for analytics based on statistics C and only a few communication updates from EDs. The importance of statistics for finding the best models rather than simply averaging them is reflected by OST + SMA, which cannot achieve low/comparable RMSE with the other variants, even if it increases significantly the communication. We obtain similar results for D1 over all predictive models; not shown for space limitations. Fig. 13 and Fig. 14 show the efficiency of our variants OST + IEAM and OST + SMA,

HOVF (Harth and Anagnostopoulos, 2017), DBP (Raza et al., 2015), INST + IEAM (Harth and Anagnostopoulos, 2018) and Global w.r.t. accuracy and communication with $\Theta \in \{50, 70, 100\}$. OST + IEAM is evidently the most efficient achieving high accuracy with least communication indicating the optimality of the OST model update rule and the IEAM model selection. DBP and HOVF are communication efficient but they do not account for the dependencies among attributes apart from selective data transfer, which has negative impact on RMSE. OST + SMA enjoys more communication efficiency than INST + IEAM but does not perform in RMSE well in terms of model selection, which is the major drawback of model averaging. INST + IEAM achieves high quality of analytics due to IEAM model selection component, but it is not as communication efficient as OST + variants due to the INST update rule, whose behavior demonstrated above. Overall, OST + IEAM is the most efficient scheme hiring the optimal OST model update rule and the IEAM error-and-input space aware model selection component dealing with models diversity.

9. Conclusions

A novel, edge-centric inferential analytics methodology is introduced contributing with time-optimized model update and diverse model selection that support analytics at the edge being communication efficient. This is achieved by optimally disseminating only knowledge/sufficient statistics instead of raw data, while the methodology introduces knowledge-driven model selection obtaining high analytics quality. Comparative assessment with baseline and schemes in the literature over real datasets evidenced its benefits in edge computing.

Our future agenda includes the vision of *data thinning*, *relevance* and *model refining* at the network edge by dramatically reducing not only the amount of data that needs to be transmitted for further processing but also the relevant data that are required for query-driven analytics tailored to analysts' and applications' needs, thus, involving *the human in the loop*.

CRedit authorship contribution statement

Christos Anagnostopoulos: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research is partially funded by the EU/H2020 GNfUV Grant #645220 and UK EPSRC Grant #EP/R018634/1.

Appendix A. Proofs

Appendix A.1. Theorem 1

Proof. The convergence of (3) involves an infinite sequence of input-error pairs $\{(\mathbf{x}, y - f_i(\mathbf{x})), \dots\}$. We adopt Robbins-Monro stochastic approximation for \mathcal{J} minimization, where the stochastic sample of \mathcal{J} decreases at each t -th input-error pair by descending in the direction of its (partial) negative gradient. Hence, by applying SGD on \mathbf{w}_k is $\Delta \mathbf{w}_k = -\frac{1}{2} \lambda \alpha_t \frac{\partial \mathcal{J}}{\partial \mathbf{w}_k}$ and on u_k is $\Delta u_k = -(1 - \lambda) \alpha_t \frac{\partial \mathcal{J}}{\partial u_k}$, where α_t satisfies $\sum_{t=0}^{\infty} \alpha_t = \infty$ and $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$ (Kosko, 1991), we obtain the update rules in Theorem 1. \square

Appendix A.2. Theorem 2

Proof. The update rule for \mathbf{w}_k based on Theorem 1 is $\Delta \mathbf{w}_k \propto (\mathbf{x} - \mathbf{w}_k)$ and let it reach equilibrium: $\Delta \mathbf{w}_k = \mathbf{0}$, given that $\|\mathbf{x} - \mathbf{w}_k\| \leq \rho_I$. We require at the convergence that each input \mathbf{x} is assigned to its winner with probability 1, i.e., $P(\|\mathbf{x} - \mathbf{w}_k\| \leq \rho_I) = 1$, which means that no other prototypes are generated.

Therefore, $P(\|\mathbf{x} - \mathbf{w}_k\| \geq \rho_I) \leq \frac{\mathbb{E}[\|\mathbf{x} - \mathbf{w}_k\|]}{\rho_I}$ based on Markov's inequality. To obtain $P(\|\mathbf{x} - \mathbf{w}_k\| \leq \rho_I) \rightarrow 1$ we have either $\rho_I \rightarrow \infty$ or $\mathbb{E}[\|\mathbf{x} - \mathbf{w}_k\|] \rightarrow 0$. Since ρ_I is a real number and interprets the concept of neighborhood, we require $\mathbb{E}[\|\mathbf{x} - \mathbf{w}_k\|] \rightarrow 0$ or $\mathbb{E}[\Delta \mathbf{w}_k] \rightarrow \mathbf{0}$. By taking the expectation of both sides we obtain:

$$\mathbf{0} = \mathbb{E}[\Delta \mathbf{w}_k] = \int_{\mathcal{X}_k} (\mathbf{x} - \mathbf{w}_k) p(\mathbf{x}) d\mathbf{x} = \int_{\mathcal{X}_k} \mathbf{x} p(\mathbf{x}) d\mathbf{x} - \mathbf{w}_k \int_{\mathcal{X}_k} p(\mathbf{x}) d\mathbf{x}.$$

This indicates that \mathbf{w}_k is constant with probability 1, and then by solving $\mathbb{E}[\Delta \mathbf{w}_k] = \mathbf{0}$, the \mathbf{w}_k equals the *centroid* $\mathbb{E}[\mathbf{x} | \mathcal{X}_k]$. Let the error u_k correspond to \mathbf{w}_k ; the median \tilde{u}_k of errors e around error prototype u_k satisfies $P(e \geq \tilde{u}_k) = P(e \leq \tilde{u}_k) = \frac{1}{2}$. Suppose that u_k has reached equilibrium, i.e., $\Delta u_k = 0$, which holds with probability 1. By taking the expectations of both sides and replacing Δu_k with the update rule $\text{sgn}(e - u_k)$ from Theorem 1:

$$\mathbb{E}[\Delta u_k] = \int \text{sgn}(e - u_k) p(e) de = P(e \geq u_k) - P(e < u_k) = 2P(e \geq u_k) - 1.$$

Since $\Delta u_k = 0$ thus u_k is constant, then $P(e \geq u_k) = \frac{1}{2}$, which denotes that u_k converges to the median of errors. \square

Appendix A.3. Lemma 1

Proof. Based on Chow and Robbins (1965), an optimal stopping rule t^* exists if it holds true for the expected return $\mathbb{E}[G_t]$ that:

- (C1) $\mathbb{E}[t^*] < \infty$
- (C2) $\mathbb{E}[G_t | \mathcal{F}_{t-1}] \geq G_{t-1}$ when $t \leq t^*$ and $\mathbb{E}[G_t | \mathcal{F}_{t-1}] \leq G_{t-1}$ when $t \geq t^*$
- (C3) $\mathbb{E}[|G_{t+1} - G_t| | \mathcal{F}_t] < M$ for all t and for some $M < \infty$.

When the above-mentioned criteria are met then $\mathbb{E}[G_{t^*}] \geq \mathbb{E}[G_t]$ where t is any stopping rule with $\mathbb{E}[t] < \infty$. In our case, C1 is clearly satisfied. Since $G_t \in [0, \Theta]$ for all t , then $\mathbb{E}[|G_{t+1} - G_t| | \mathcal{F}_t] \leq 2\Theta$, thus C3 is satisfied. For criterion C2 we have that: for $S_t > \Theta$, C2 clearly holds. We have for $S_t < \Theta$,

$$\mathbb{E}[G_{t+1} | \mathcal{F}_t] = S_t F(\Theta - S_t) + \int_0^{\Theta - S_t} z dF(z),$$

and

$$G_t - \mathbb{E}[G_{t+1} | \mathcal{F}_t] = S_t(1 - F(\Theta - S_t)) - \int_0^{\Theta - S_t} z dF(z).$$

The right hand side of the second expression is non-decreasing thus criterion C2 holds true. \square

Appendix A.4. Theorem 4

Proof. Given that $S_t \leq \Theta$, the conditional expectation $\mathbb{E}[G_{t+1}(S_{t+1}) | S_t \leq \Theta]$ is given by

$$\begin{aligned} \mathbb{E}[G_{t+1}(S_{t+1}) | S_t \leq \Theta] &= \mathbb{E}_Z[G_{t+1}(S_t + Z) | S_t \leq \Theta, S_t + Z \leq \Theta] P(S_t + Z \leq \Theta) + \mathbb{E}_Z \\ &\quad [G_{t+1}(S_t + Z) | S_t \leq \Theta, S_t + Z > \Theta] P(S_t + Z > \Theta) = \mathbb{E}_Z[S_t + Z | Z \leq \Theta - S_t] P(Z \leq \Theta - S_t) = S_t F_Z(\Theta - S_t) + \int_0^{\Theta - S_t} z dF_Z(z). \end{aligned}$$

Note that, based on the definition of $G_t(S_t)$ in (6), G_{t+1} is zero for $S_{t+1} = S_t + Z > \Theta$, thus, the conditional expectation $\mathbb{E}_Z[G_{t+1}(S_t + Z) | S_t \leq \Theta, S_t + Z > \Theta] = 0$. The mechanism stops at the first time instance t with S_t such that $\mathbb{E}[G_{t+1}(S_{t+1}) | S_t \leq \Theta] \leq S_t$. The corresponding difference is monotonically non-increasing with S_t with $S_t < \Theta$, thus, the 1-sla rule is optimal. \square

Appendix A.5. Lemma 2

Proof. From Theorem 4, the optimal stopping rule exists $t^* < \infty$ if $F_Z(\Theta) < 1$ given that at time $t = 1$, the ED experiences $S_1 = 0$. In this case, the ED decides to take another observation of the discrepancy Z . Otherwise, if $F_Z(\Theta) \geq 1$ then the stopping time is infinite and there will be never the case the conditional $\mathbb{E}[G_{t+1} | \mathcal{F}_t]$ be less than G_t . Hence, the ED decides not to take any other discrepancy observation, thus, no delay,

and immediately updates the model to the EG. Based on Markov inequality and $Z > 0$ by definition, we obtain that $F_Z(\Theta) > 1 - \frac{\mathbb{E}[Z]}{\Theta}$, thus, for Θ it must hold true that $F_Z(\Theta) \in (1 - \frac{\mathbb{E}[Z]}{\Theta}, 1)$ and then $\mathbb{E}[Z] < \Theta$. \square

Appendix A.6. Proposition 1

Proof. Without loss of generality, regression error residuals of f and f^p are Gaussian distributions with *unknown* deviations σ, σ_0 , respectively (Nolan and Ojeda-Revah, 2013; Blattberg and Sargent, 1971). Then, Z follows the folded Gaussian with variance $\sigma_Z^2 = (\sigma^2 + \sigma_0^2)(1 - \frac{2}{\pi})$ and median $m_Z = \sqrt{2(\sigma^2 + \sigma_0^2)}\text{erf}^{-1}(1/2)$; $\text{erf}^{-1}(x)$ is the inverse of the error function $\text{erf}(x)$. We obtain $P\{Z > \theta\} = 1 - P\{Z \leq \theta\} = 1 - F_Z(\theta)$ with $F_Z(z) = \text{erf}(\frac{z}{\sqrt{2(\sigma^2 + \sigma_0^2)}})$ being the cumulative distribution function of Z . Hence:

$$\mathbb{E}[M] = TP\{Z > \theta\} = T(1 - F_Z(\gamma m_Z)) = T(1 - \text{erf}(\gamma \cdot \text{erf}^{-1}(0.5)))$$

for $\gamma = 1$ we obtain $\mathbb{E}[M] = T/2$ as expected. \square

Appendix A.7. Proposition 2

Proof. Consider the k -th dimension change indicator $I(\Delta w_{k,\ell}) = 1$, if k -th dimension of the closest \mathbf{w}_ℓ significantly changes, i.e., $\Delta w_{k,\ell} > \epsilon$, due to the update rule upon a pair (\mathbf{x}, y) ; 0 otherwise, for any arbitrary $\epsilon > 0$. We define the indicator $I(\Delta u_\ell)$ in a similar way. The probability of changing k out of $d + 1$ dimensions of the change vector $\Delta(\mathbf{w}_\ell, u_\ell)$ based on the corresponding indicators is $\left(\frac{d+1}{k}\right)^{-1}$, $k = 0, \dots, d + 1$. Given that the ℓ -th input/error prototype is adapted (Algorithm 1), the expected magnitude of change is then:

$$\begin{aligned} \mathbb{E}[\|\Delta \mathbf{w}_\ell, u_\ell\|] &= \sum_{k=0}^{d+1} \left(\frac{d+1}{k}\right)^{-1} (\lambda \alpha \|\mathbf{x} - \mathbf{w}_\ell\|, (1 - \lambda) \alpha \text{sgn}(e - u_\ell)) \leq \sum_{k=0}^{d+1} \left(\frac{d+1}{k}\right)^{-1} (\lambda \alpha \rho_I, (1 - \lambda) \alpha \rho_O) \\ &= \alpha F_{d+1}^{(0)} \max(\lambda \rho_I, (1 - \lambda) \rho_O) \% 2\alpha(1 + d^{-1} + 2^{-(d+1)}) \max(\lambda \rho_I, (1 - \lambda) \rho_O) \end{aligned}$$

where $F_x^{(y)} = \sum_{k=0}^x k^y \left(\frac{x}{k}\right)^{-1}$ is the sum involving the y moments of the reciprocals of binomial coefficients (Belbachir et al., 2011) with asymptotic expansion $F_x^{(0)} \% 2 + 2(x - 1)^{-1} - 2^{1-x}$ (Yanget al, 2010). \square

Appendix A.8. Theorem 5

Proof. If there is a polynomial-time algorithm for MSAP, then a polynomial-time algorithm can be developed for SSP. Assume there exists a polynomial algorithm $A(\mathcal{F}, s)$ that solves MSAP, i.e., $A(\mathcal{F}, s)$ finds in polynomial time the minimum subset \mathcal{F}' subject to constraint in Problem 2. Then, $A(\mathcal{F}, s)$ can be used to solve SSP with (\mathcal{F}, ns) , $n = |\mathcal{F}|$. In general, any solution $B(\mathcal{F}, s)$ of SSP with (\mathcal{F}, s) can be formulated as shown in Algorithm 4. If the complexity of $A(\mathcal{F}, s)$ is a polynomial $\mathcal{Q}(n)$ then the complexity of $B(\mathcal{F}, s)$ is $O(n\mathcal{Q}(n))$. But, this implies that there is a polynomial-time algorithm for SSP. Hence, no polynomial-time algorithm exists for MSAP.

Algorithm 3 $B(\mathcal{F}, s)$.

Input: \mathcal{F}, s
Output: \mathcal{F}'
1: **for** $1 \leq k \leq |\mathcal{F}|$ **do**
2: **call** $A(\mathcal{F}, \frac{s}{k})$
3: **If** a subset \mathcal{F}' of \mathcal{F} with k elements is found, whose elements have an average k' such that $\lfloor k' \rfloor = s/k$ or $\lceil k' \rceil = s/k$ **Then** return \mathcal{F}' .
4: **end for**

\square

Appendix A.9. Corollary 1

Proof. Consider the errors $e = |\hat{y} - y|$ and $e' = |\hat{y}' - y|$. In order to show that the problem of finding the minimum subset \mathcal{F}' with $e' = e$ is NP-hard, it suffices to show that finding the minimum subset $\mathcal{F}' \subset \mathcal{F}$ of cached models such that $\hat{y} = \hat{y}'$ subject to constraint in Problem 2 is NP-hard. Consider the set $\mathcal{F}^0 = \{|\hat{y}_i|\}_{i=1}^n$, and $\mathcal{F}^1 = \{|\hat{y}_i|\}_{i=1}^n$ with $\hat{y}_i > 0, \forall i$. Since MSAP, which deals with integers is NP-hard from Theorem 3, MSAP with $(\mathcal{F}^0, |\hat{y}|)$ and $(\mathcal{F}^1, |\hat{y}|)$ is also NP-hard. \square

Appendix B. Predictive models adaptation

LM and AR-RLS Incremental Model Update: The ED i expresses x_t as a function of its previous values plus noise ϵ_t , i.e., $x_t = \varphi_1 x_{t-1} + \dots + \varphi_N x_{t-N} + \epsilon_t$, where the window length N is determined by the Akaike Information Criterion (AIC) that penalizes model complexity (Burnham and Anderson, 2002). To estimate the coefficients φ , the ED i adopts the Recursive Least Squares (RLS) that allows dynamic update of a least-squares fit, as used in the LM. The least-squares solution to a system of equations $\mathbf{A}\mathbf{b} = \mathbf{y}$, where $\mathbf{A} \in \mathbb{R}^{N \times d-1}$, $\mathbf{y} \in \mathbb{R}^N$ and $\mathbf{b} \in \mathbb{R}^d$ regression coefficients to be estimated is given by the solution of $\mathbf{A}^\top \mathbf{A}\mathbf{b} = \mathbf{A}^\top \mathbf{y}$. Hence, we need only the projections: $\mathbf{P} = \mathbf{A}^\top \mathbf{A}$ and $\mathbf{p} = \mathbf{A}^\top \mathbf{y}$, requiring $O(d^2)$ space to keep the model up to date. When a new embedding vector arrives with y_{N+1} and \mathbf{x}_{N+1} , we update $\mathbf{P} \leftarrow \mathbf{P} + \mathbf{x}_{N+1} \mathbf{x}_{N+1}^\top$ and

$\mathbf{p} \leftarrow \mathbf{p} + y_{N+1} \mathbf{x}_{N+1}$. The coefficients \mathbf{b} are incrementally updated in $O(d)$ as:

$$\mathbf{G} \leftarrow \mathbf{G} - (1 + \mathbf{x}_{N+1}^\top \mathbf{G} \mathbf{x}_{N+1})^{-1} \mathbf{G} \mathbf{x}_{N+1} \mathbf{x}_{N+1}^\top \mathbf{G} \quad (\text{B.1})$$

$$\mathbf{b} \leftarrow \mathbf{b} - \mathbf{G} \mathbf{x}_{N+1} (\mathbf{x}_{N+1}^\top \mathbf{b} - y_{N+1}) \quad (\text{B.2})$$

where \mathbf{G} is initialized to $\epsilon \mathbf{I}$ with $\epsilon > 0$ and \mathbf{I} is the $d \times d$ identity matrix. In the case of AR-RLS the solution vector \mathbf{b} consists precisely of the AR coefficients φ , i.e., $\mathbf{b} = [\varphi_1, \dots, \varphi_d]^\top \in \mathbb{R}^d$.

RBF and AR-RBF Incremental Model Update: The ED i adopts a computationally efficient incremental learning methodology (Schwenker et al., 2001) of the RBF parameters upon reception of a pair (\mathbf{x}, y) , in case of regression, or embedding vector \mathbf{u} , in case of time series. Based on this methodology, with update complexity $O(dM)$ over M neuron, the incremental update rules with updating rate $\alpha \in (0, 1)$ for centers μ_{im} and weights a_{im} are:

$$\Delta a_{im} \leftarrow \alpha h(\|\mathbf{x} - \mu_{im}\|)(y - f_i(\mathbf{x})) \quad (\text{B.3})$$

$$\Delta \mu_{im} \leftarrow \alpha h(\|\mathbf{x} - \mu_{im}\|)(y - f_i(\mathbf{x})) a_{im} (\mathbf{x} - \mu_{im}) \quad (\text{B.4})$$

We obtain the time-series coefficients for AR-RBF by replacing in (B.3) and (B.4) $\mathbf{x} = [u_{t-1}, \dots, u_{t-d+1}]$ and $y = u_t$ from the embedded vector \mathbf{u} , $m \in [M]$.

Appendix C. Upper-bound Probability of Model Adaptation

Based on the Algorithm 1 (Line 15), the ED i decides on adapting the local model f_i given a classified *familiar* input-output pair (\mathbf{x}, y) , should the prediction error exceeds the (adjusted) error tolerance threshold ρ_0 . This event, coined here as $A = \{\text{ModelAdaptation}\}$ occurs with probability $P\{A\} = P(\|\mathbf{x} - \mathbf{w}_k\| > \rho_l)P(e > \rho_0)$, given that \mathbf{w}_k is the closest prototype to input \mathbf{x} , i.e., the pair is classified as not familiar given a familiarization threshold ρ_l and the model predictability error exceeds ρ_0 . Let $F_e(x) = P(e \leq x)$ be the cumulative distribution function of the model prediction error e . Then, $P\{A\} = (1 - P(\|\mathbf{x} - \mathbf{w}_k\| \leq \rho_l))(1 - P(e \leq \rho_0)) = (1 - F_e(\rho_0))(1 - P(\|\mathbf{x} - \mathbf{w}_k\| \leq \rho_l))$. Based on the Markov's inequality, we obtain that $P\{A\} \leq (1 - F_e(\rho_0)) \frac{E[\|\mathbf{x} - \mathbf{w}_k\|]}{\rho_l}$, which is the upper bound of the model adaptation probability.

In convergence of the input-error prototypes we obtain that $E[\|\mathbf{x} - \mathbf{w}_k\|] \rightarrow 0$, which derives from the fact that $P(\|\mathbf{x} - \mathbf{w}_k\| \leq \rho_l) \rightarrow 1$, as proved in Appendix A.2. Hence, in this case, the upper bound model adaptation probability tends to zero, indicating that the rate of adaptation *decreases* as long as the input-error vectorial space is quantized to minimize both the EQE and EPE. In this stage, the model does not require adaptation, which is anticipated since the ED i classifies the input-output pairs as familiar. This indicates the capability of the Algorithm 1 to ensure minimization of the expected computational overhead on the EDs via the proposed hetero-associative inference mechanism. However, in a dynamic environment, there are anticipated concept drifts over the input-output distribution. Hence, the upper probability of model adaptation can be envisaged as the model adaptation rate in a specific time horizon. As a candidate mechanism to handle this context is to monitor a descriptive statistic of the most recent familiar input-output pairs, e.g., a moving centroid of the N_i recent familiar (\mathbf{x}, y) pairs received in ED i . Given that there exists a significant change in this descriptive statistic (e.g., adopting the CuSum algorithm (Basseville and Nikiforov, 1993)), the local model can be triggered to adapt/re-trained based these recent familiar input-output vectors, which is expected to reduce the inherent complexity to be adapted on every familiar pair received. This is left in our future research agenda.

Appendix D. Supplementary data

Supplementary data related to this article can be found at <https://doi.org/10.1016/j.jnca.2020.102696>.

References

- Alghamdi, I., Anagnostopoulos, C., Pezaros, D.P., 2019a. Delaytolerant sequential decision making for task offloading in mobile edge computing environments. *Information* 10, 312.
- Alghamdi, I., Anagnostopoulos, C., Pezaros, D.P., 2019b. On the optimality of task offloading in mobile edge computing environments. In: 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, pp. 1–6.
- Alghamdi, I., Anagnostopoulos, C., Pezaros, D.P., 2019c. Time-optimized task offloading decision making in mobile edge computing. In: 2019 Wireless Days (WD), Manchester, United Kingdom, pp. 1–8.
- Anagnostopoulos, C., Triantafillou, P., June 2017a. Query-driven learning for predictive analytics of data subspace cardinality. *ACM Trans. Knowl. Discov. Data* 11 (4), 46 Article 47.
- Anagnostopoulos, C., Triantafillou, P., 2017b. Efficient scalable accurate regression queries in in-DBMS analytics. In: IEEE 33rd International Conference on Data Engineering (ICDE) 2017, CA, pp. 559–570.
- Anagnostopoulos, C., Savva, F., Triantafillou, P., 2018. Scalable aggregation predictive analytics: a query-driven machine learning approach. *Appl. Intell.* 48 (9), 2546–2567.
- Anand, N., Chintalapally, A., Puri, C., Tung, T., 2017. Practical Edge Analytics: Architectural Approach and Use Cases. In: IEEE EDGE 2017, pp. 236–239.
- Babcock, B., Datar, M., Motwani, R., 2002. Sampling from a moving window over streaming data. In: Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '02). Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, pp. 633–634.
- Basseville, M., Nikiforov, I.V., 1993. Detection of Abrupt Changes: Theory and Application. Prentice-Hall, Englewood Cliffs, NJ.
- Belbachir, H., Rahmani, M., Sury, B., 2011. Sums involving moments of reciprocals of binomial coefficients. *J. Integer Seq.* 14 (6), 16 Article 11.6.6.
- Bilal, K., Khalid, O., Erbad, A., Khan, S.U., 2018. Potentials, trends, and prospects in edge technologies: fog, cloudlet, mobile edge, and micro data centers. *Comput. Network.* 130, 94–120.
- Blattberg, R., Sargent, T., 1971. Regression with non-Gaussian stable disturbances: some sampling results. *Econometrica* 39 (3), 501–510.
- Bottou, L., Bousquet, O., 2007. The tradeoffs of large scale learning. In: NIPS07, pp. 161–168.
- Brown, G., Wyatt, J.L., Tino, P., Dec 2005. Managing diversity in regression ensembles. *J. Mach. Learn. Res.* 6, 1621–1650.
- Bruss, F.T., Le Cam, L. (Eds.), 2000. Game Theory, Optimal Stopping, Probability and Statistics: Papers in Honor of Thomas S. Ferguson. Institute of Mathematical Statistics, Beachwood, Ohio.
- Burnham, K., Anderson, D., 2002. Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach, second ed. Springer-Verlag.
- Carpenter, G.A., Grossberg, S., March 1988. The ART of adaptive pattern recognition by a self-organizing neural network. *Computer* 21 (3), 77–88.
- Cerqueira, V., Torgo, L., Oliveira, M., Pfahringer, B., 2017. Dynamic and heterogeneous ensembles for time series forecasting. In: IEEE International Conference on Data Science and Advanced Analytics (DSAA), Tokyo, pp. 242–251.
- Chow, Y., Robbins, H., 1965. On optimal stopping rules for S_n/n . *Ill J. Math.* 9 (3), 444–454.
- Cormode, G., May 2013. The continuous distributed monitoring model. *ACM SIGMOD Rec* 42 (1), 5–14.
- Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y., 2006. Online passive-aggressive algorithms. *J. Mach. Learn. Res.* 7 (December 2006), 551–585.

- Engel, Y., Mannor, S., Meir, R., Aug 2004. The kernel recursive least-squares algorithm. *IEEE Trans. Signal Process.* 52 (8), 2275–2285.
- Ferreira, P.M., Ruano, A.E., Sept. 2009. Online sliding-window methods for process model adaptation. *IEEE TIM* 58 (9), 3012–3020.
- Gabel, M., Keren, D., Schuster, A., 2015. Monitoring least squares models of distributed streams. In: *ACM KDD* 2015, NY, pp. 319–328.
- Garcia Lopez, P., et al., 2015. Edgecentric computing: vision and challenges. *ACM SIGCOMM Comput. Commun. Rev.* 45 (5), 37–42.
- Garey, M.R., Johnson, D.S., 1990. *Computers and Intractability; a Guide to the Theory of NpCompleteness*. W. H. Freeman & Co., New York, NY, USA.
- Giang, N.K., et al., 2015. Developing IoT applications in the fog: a distributed dataflow approach. In: *IEEE IOT* 2015, Seoul, pp. 155–162.
- Harth, N., Anagnostopoulos, C., 2017. Quality-aware aggregation & predictive analytics at the edge. In: *IEEE International Conference on Big Data (Big Data)*, MA, pp. 17–26.
- Harth, N., Anagnostopoulos, C., Jul 2018. Edge-centric efficient regression analytics. In: *IEEE International Conference on Edge Computing (EDGE)*, San Francisco, CA, USA.
- Harth, N., Anagnostopoulos, C., Pezaros, D., 2017. Predictive intelligence to the edge: impact on edge analytics. *Evolving Syst.* 8 (124).
- Idreos, S., Papaemmanouil, O., Chaudhuri, S., 2015. Overview of data exploration techniques. In: *ACM SIGMOD International Conference on Management of Data (SIGMOD 15)*. Association for Computing Machinery, New York, NY, USA, pp. 277–281.
- Jain, R., Tata, S., 2017. Cloud to edge: distributed deployment of process-aware IoT applications. In: *IEEE EDGE* 2017, pp. 182–189.
- Kamath, G., Agnihotri, P., Valero, M., Sarker, K., Song, W.Z., 2016. Pushing analytics to the edge. In: *IEEE Global Communications Conference (GLOBECOM)*, Washington, DC, pp. 1–6.
- Kaneda, Y., Mineno, H., 2016. Sliding window-based support vector regression for predicting micrometeorological data. *ESWA J.* 59, 217–225.
- Kohonen, T., Schroeder, M.R., Huang, T.S., 2001. *SelfOrganizing Maps*, third ed. Springer-Verlag, Berlin, Heidelberg.
- Koneny, J., McMahan, H.B., Yu, F., Richtarik, P., Suresh, A.T., Bacon, D., 2016. Federated learning: strategies for improving communication efficiency. In: *NIPS 2016, Private Multi-Party Machine Learning Workshop*.
- Kosko, B., Sep 1991. Stochastic competitive learning. *IEEE TNN* 2 (5), 522–529.
- Lazerson, A., Keren, D., Schuster, A., 2016. Lightweight monitoring of distributed streams. In: *ACM KDD* 2016, NY, pp. 1685–1694.
- Liu, Z., Heer, J., 31 Dec. 2014. The effects of interactive latency on exploratory visual analysis. *IEEE Trans. Visual. Comput. Graph.* 20 (12), 2122–2131.
- Liu, Y., Quan, W., Wang, T., Wang, Y., Oct. 2018. Delay-constrained utility maximization for video Ads push in mobile opportunistic D2D networks. *IEEE Internet Things J.* 5 (5), 4088–4099.
- Liu, Y., Wang, H., Peng, M., Guan, J., Xu, J., Wang, Y., Dec 2019a. DeePGA: a privacy-preserving data aggregation game in crowdsensing via deep reinforcement learning. *IEEE Internet Things J.*, <https://doi.org/10.1109/JIOT.2019.2957400>.
- Liu, Y., Hao, L., Liu, Z., Sharif, K., Wang, Y., Das, S.K., July 2019b. Mitigating interference via power control for two-tier femtocell networks: a hierarchical game approach. *IEEE Trans. Veh. Technol.* 68 (7), 7194–7198.
- Lukaszky, S., 2004. A new concept of probability metric and its applications in approximation of scattered data sets. *Comput. Mech.* 33, 299–304.
- Mendes-Moreira, J., Soares, C., Jorge, A.M., Freire De Sousa, J., Dec. 2012. Ensemble approaches for regression: a survey. *ACM Comput. Surv.* 45 (1) article 10.
- Nolan, J.P., Ojeda-Revah, D., 2013. Linear and nonlinear regression with stable errors. *J. Econom.* 172 (2), 186–194 Elsevier.
- Przydatek, B., Jul 2002. A fast approximation algorithm for the subset sum problem. *Int. Trans. Oper. Res.* 9 (4), 437–459.
- Raza, U., Camerra, A., Murphy, A.L., Palpanas, T., Picco, G.P., Aug. 1 2015. Practical data prediction for real-world wireless sensor networks. *IEEE TKDE* 27 (8), 2231–2244.
- Renart, E.G., Diaz-Montes, J., Parashar, M., 2017. Data-driven stream processing at the edge. In: *IEEE ICPEC* 2017, pp. 31–40.
- Savva, F., Anagnostopoulos, C., Triantafyllou, P., 2018. Explaining aggregates for exploratory analytics. In: *2018 IEEE International Conference on Big Data (Big Data)*, Seattle, WA, USA, pp. 478–487.
- Savva, F., Anagnostopoulos, C., Triantafyllou, P., 2019. Aggregate query prediction under dynamic workloads. In: *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA, pp. 671–676.
- Savva, F., Anagnostopoulos, C., Triantafyllou, P., 2020. SuRF: identification of interesting data regions with surrogate models. In: *36th IEEE International Conference on Data Engineering (IEEE ICDE)*, Dallas, TX, USA April 2020.
- Schwenker, F., Kestler, H.A., Palm, G., May 2001. Three learning phases for radial-basis-function networks. *Neural Network.* 14 (45), 439–458.
- Sharma, S.K., Wang, X., 2017. Live data analytics with collaborative edge and Cloud processing in wireless IoT networks. *IEEE Access* 5, 4621–4635.
- Shen, F., Hasegawa, O., June 2006. An adaptive incremental LBG for vector quantization. *Neural Network.* 19 (5), 694–704.
- Shiryayev, A.N., 2008. first ed. *Optimal Stopping Rules, Stochastic Modelling and Applied Probability*, vol. 8. Springer-Verlag Berlin Heidelberg.
- Takens, F., 1981. Dynamical systems and turbulence. In: Rand, D.A., Young, L.-S. (Eds.), *Warwick 1980: Proceedings of a Symposium Held at the University of Warwick 1979/80. Detecting strange attractors in turbulence*, pp. 366–381. ch.
- Tatbul, N., Cetintemel, U., Zdonik, S., Cherniack, M., Stonebraker, M., 2003. Load shedding in a data stream manager. In: *29th International Conference on Very Large Data Bases (VLDB 03)*, vol. 29. VLDB Endowment, pp. 309–320.
- Trevor, H., Tibshirani, R., Friedman, J., 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, second ed. Springer.
- Wang, H., Li, C., 2016. Distributed quantile regression over sensor networks. *IEEE Trans. Signal Inf. Process. Netw.* 99.
- Wei, X., et al., 2017. MVR: an architecture for computation offloading in mobile edge computing. In: *IEEE EDGE* 2017, Honolulu, pp. 232–235.
- Yang, J.-H., et al., 2010. The asymptotic expansions of certain sums involving inverse of binomial coefficient. *Int. Math. Forum* 5 (16), 761–768.
- Zhou, ZhiHua, 2012. *Ensemble Methods: Foundations and Algorithms*, first ed. Chapman & Hall, CRC.

Dr Christos (Chris) Anagnostopoulos is a Lecturer (Assistant Professor) in Distributed and Pervasive Computing in the School of Computing Science at the University of Glasgow. His expertise is in the areas of network-centric pervasive & adaptive systems and in-network information processing in large-scale distributed computing networks. He has received funding for his research by the EC/H2020, UK EPSRC, and the industry. Dr Anagnostopoulos is coordinating (Principal Investigator) the projects: EU H2020/GNFUV and EU H2020 Marie Skłodowska-Curie (MSCA)/INNOVATE, and is a co-PI of the EU PRIMES and UK EPSRC CLDS. Dr Anagnostopoulos is an author of over 140 refereed scientific journals/conferences. He is leading the Essence: Pervasive & Distributed Intelligence within the Knowledge and Data Engineering Systems Group (IDA Research Section). He is also an associate member of the Networked Systems Research (NETLAB). Dr Anagnostopoulos before joining Glasgow was an Assistant Professor at Ionian University and Adjunct Assistant Professor at the University of Athens and University of Thessaly. He has held postdoctoral positions at University of Glasgow and University of Athens in the areas of in-network context-aware mobile computing systems. He holds a BSc, MSc, and PhD in Computing Science, University of Athens. He is an associate fellow of the HEA, member of ACM and IEEE.